I. P. Sharp Associates

WSDOC User's Guide

WSDOC

SHARP APL WORKSPACE DOCUMENTATION FACILITY

USER'S GUIDE

PREFACE

This book describes the SHARP APL Workspace Documentation Facility: WSDOC. Specifically, this User's Guide describes how you produce a WSDOC report for a workspace; describes the WSDOC state options in detail; shows a sample WSDOC; describes how WSDOC's source files are organized; and describes how you can set up a WSDOC profile.

You should find this book useful if you write programs in SHARP APL, or if you want to produce hardcopy reports of existing SHARP APL workspaces. You should be familiar with the SHARP APL system and language, and their principal features.

If ycu want more information about WSDOC beyond what is contained in this book, contact your I. P. Sharp representative. If you're enrolled in the I. P. Sharp MAILBOX, you can send questions or comments about WSDOC to address code *SYSL*.

History

WSDOC was developed and implemented in 1974 by Leslie H. Goldsmith, and enhanced extensively in 1978.

Printed in Canada
September 1984
Publication code: 0349 8409 E1
ISBN 0 86493 094 1

CONTENTS

1. WHAT IS WSDOC? 1
   - Who Uses WSDOC? 1
   - Where Is WSDOC? 1

2. HOW TO USE WSDOC 2
   - Creating the Source File: The *WSTOFILE* Program 2
   - Managing the WSDOC State: The *STATE* Workspace 3
   - Executing the *WSDOC* Program 4
   - Documenting Several Workspaces at Once 5

3. THE WSDOC STATE 7
   - What Is the State? 7
   - Displaying the State 7
   - The State Options 8
   - The Default State 9
   - Changing the State 9
   - Putting the State to File 9
   - Getting the State from File 10

4. WSDOC CONTROL OPTIONS 11
   - The *TTASK*, *NTASK*, and *BTASK* Options 11
   - The *PRINT* and *NOPRINT* Options 12
   - The *KEYB* Option 12
   - The *PAGEPAUSE* Option 12

5. WSDOC FILE OPTIONS 13
   - The *ERASESRC* and *SAVESRC* Options 13
   - The *OUTFILE* and *CREATEOUT* Options 13
   - The *HIGHSPEED* and *NOHIGHSPEED* Options 14
   - The *HSPRINT* Option 14

6. WSDOC FORMAT OPTIONS 16
   - Page Format 16
   - The *WIDTH* Option 18
   - The *DEPTH* Option 18
   - The *SET MARGINS* Option 18
   - The *TITLE* Option 18
   - The *PAGENOS* Option 19
   - The *TREEHEIGHT* Option 20
   - The *PRINTSUMS* Option 20
   - The *ALPHASORT* Option 20
   - The *PAGEFORMAT* Option 21
   - The *STAMP* and *NOSTAMP* Options 22
   - The *DIAMOND* and *NODIAMOND* Options 22
   - Alternate Definition Formats 23

7. WSDOC CONTENT OPTIONS 24
   - The *WSENV* Option 24
   - The *LIST*, *NOLIST*, *LISTONLY*, and *NOLISTONLY* Options 25

The *ONLYFNDEF* and *NOFNDEF* Options 26
The *ONLYVARDEF* and *NOVARDEF* Options 26
The *ONLYPKGDEF* and *NOPKGDEF* Options 27
The *QUOTES* and *NOQUOTES* Options 27
The *QUADFNS* and *NOQUADFNS* Options 27
The *BRANCHFNS* Option 27
The *TREEROOTS* Option 28
The *SURROGATES* Option 28
The *XREFSYMBS* Option 28

8. WSDOC REPORT ELEMENTS 29
The *BANNERS* Option 30
The *COVER* Option 30
The *TBLCONT* Option 30
The *FNS* Option 30
The *VARS* Option 30
The *GRPS* Option 31
The *GRP* Option 31
The *FNSUM* and *FNSUMC* Options 31
The *VARSUM* Option 33
The *GRPSUM* Option 33
The *PKGSUM* Option 33
The *FNDEF* Option 34
The *VARDEF* Option 34
The *PKGDEF* Option 35
The *FNXREF* Option 36
The *WSXREF* Option 37
The *FNCALLS* Option 39
The *WSGRAPH* Option 39
The *MSGS* Option 40
The *SYSCMDS*, *SUMS*, *DEFS*, and *XREFS* Options 40
The *ALL* and *NONE* Options 41

9. WSDOC PROFILES 42

10. WSDOC UTILITY PROGRAMS 44
The *GENWSTOFILE* Program 44
The *MRGWSTOFILES* Program 45
The *WSDOCSTATUS* Program 45

APPENDICES
A. WSDOC Source File Organization 46
B. WSDOC State Control Vectors 47
C. WSDOC State Option Summary 49
D. WSDOC Glossary 54

INDEX 56

Chapter 1

WHAT IS WSDOC?

The SHARP APL Workspace Documentation Facility is used to analyse the contents and environment of a SHARP APL workspace, and to produce a report of that analysis.

WSDOC provides:

- the names of functions, variables, groups, and packages in the workspace;
- formatted listings of these objects;
- cross-reference tables that show where and how identifiers are used in the workspace; and
- tree structure diagrams that illustrate the hierarchical calling structure of programs.

WSDOC can produce its output on a terminal, or put it to file. A WSDOC output file can be printed at an I. P. Sharp office.

Because the workspace you want to document may be nearly full, WSDOC operates by first creating a **source file** that contains an image of the workspace to be documented. WSDOC then uses this workspace image as its input.

WSDOC output is available in various formats and in several levels of detail. The format and detail are determined by a set of parameters collectively called the **state**. These parameters have initial settings which together form the **default state**. The default state remains in effect until you explicitly reset it.

Who Uses WSDOC?

WSDOC is useful for those who:

- write programs or systems in SHARP APL; or
- want to analyse and report a workspace in order to understand or use it better.

A WSDOC report is valuable when a system is being debugged, and the report can form an important part of the system's permanent documentation. When you're maintaining a system (especially one you didn't write), a WSDOC report can be your first step toward understanding the system's inner workings.

Where Is WSDOC?

WSDOC can be found in the following workspaces in library 7:

```
7 WSTOFILE
7 STATE
7 WSDOC
```

Workspace `7 WSTOFILE` contains only the *WSTOFILE* program, which is used to create and fill a file with an image of the active workspace, and to load `7 STATE`.

Workspace `7 STATE` contains WSDOC's state-setting functions. It also contains the online documentation for WSDOC, in variables *DESCRIBE* and *WSDOCHOW*.

Workspace `7 WSDOC` contains the *WSDOC* program itself.

Chapter 2

HOW TO USE WSDOC

There are three steps involved in using WSDOC:

1 The *WSTOFILE* program is used to create a **source file** that contains an image of the workspace to be documented.
2 The state-setting functions are used to change the WSDOC **state**, and to put the changed state to the source file.
3 The *WSDOC* program itself is used to analyse the workspace image in the source file and to produce a report of this analysis—on a terminal, or spooled to an **output file**.

You need not go through all three stages in sequence for every workspace you want to document: a changed WSDOC state can be applied to several workspaces; and the *WSDOC* program can take several workspace images for input at the same time.

I. P. Sharp's *HSPRINT* facility allows you to produce a WSDOC source file at an I. P. Sharp office. Using *HSPRINT* to produce a long WSDOC report is less expensive (and can be more convenient) than printing it on your terminal. WSDOC can submit an *HSPRINT* request for you.

WSDOC's auxiliary programs are used to merge WSDOC source files, or to incorporate program files into a WSDOC source file.

Creating the Source File: The *WSTOFILE* Program

The workspace to be documented with WSDOC may be nearly full. WSDOC avoids filling the workspace with itself by creating an **image** of the workspace, and putting this image to a source file. WSDOC actually documents this image, rather than the original workspace.

The *WSTOFILE* program is used to make a WSDOC source file.

Use *WSTOFILE* as follows:

1 Load the workspace you want to document.
2 Make sure that the active workspace contains no extraneous objects (for example, a function you copied from another workspace after you loaded the active workspace).
3 Copy the *WSTOFILE* workspace into the active workspace (the *WSTOFILE* workspace contains only the function *WSTOFILE*).
4 Execute the *WSTOFILE* program.

WSTOFILE tries to create a source file with the same name as the active workspace. If a file with that name already exists, *WSTOFILE* prompts for another name. You must then enter a name, or else enter a space followed by a carriage return (SPACE RETURN) to exit. (You might want to exit to erase the file with the workspace's name, and then re-execute *WSTOFILE*.)

WSTOFILE creates the source file (whatever its name) and ties it to the lowest available tie number greater than or equal to 20.

After it has created the source file, *WSTOFILE*:

- Displays the source file's name and tie number;
- Displays the number of functions, variables (including packages), and groups in the active workspace—except *WSTOFILE* itself;
- Appends images of the workspace's environment and objects to the source file;
- Stores the default WSDOC state in the source file; and
- Quietly loads the *STATE* workspace.

Workspace 7 *STATE* contains an unlocked and commented version of *WSTOFILE*, with the name *WSTOFILE*.

Example. Let's suppose you have a workspace named *UTILITY* that you'd like to document:

```
      )LOAD UTILITY
SAVED  15.22.07 05/25/84
      )COPY 7 WSTOFILE
SAVED  20.30.31 03/18/84

      WSTOFILE
FILE NAME - 1234567 UTILITY - IN USE.
ALTERNATE FILE NAME OR SPACE,CR:  SPACE RETURN

      'UTILITY' ⎕STIE 60
      'UTILITY' ⎕ERASE 60

      WSTOFILE
SRCFILE 20 - 1234567 UTILITY
33 FUNCTIONS
18 VARIABLES
4 GROUPS
      )WSID
IS 7 STATE
```

Managing the WSDOC State: The *STATE* Workspace

The *STATE* workspace contains functions you can use to display the WSDOC state, to reset the state, and to store the state in WSDOC source files.

When you load 7 *STATE*, your *STATE* profile is sought and executed; if you don't have such a profile, the default *STATE* profile is executed. The default profile calls the *WSDOCSTATUS* program (which is described in "WSDOC Utility Programs"). A *STATE* profile typically displays the status of previous WSDOCs, and resets certain state options to the values you usually prefer. For information about how to set or reset a *STATE* profile, see "WSDOC Profiles".

You can display the current state at any time with the *STATE* function. When you load 7 *STATE* the first time, the **default state** is in effect. You can restore this state at any time by executing the *DEFAULT* program. You reset state options with **state-setting functions**.

When you're satisfied with the state, you use the *PUTSTATE* function to store it in the appropriate source files. *PUTSTATE*'s right argument is a single source file's name or any number of source files' tie numbers.

If your session parameter (system variable ⎕SP) is an empty vector (its default value) or a numeric vector, *PUTSTATE* also puts the valid source file tie numbers into ⎕SP. This way, the tie numbers can be passed to the *WSDOC* program in workspace 7 *WSDOC*.

Example. We now continue documenting your *UTILITY* workspace:

```
      )WSID
IS 7 STATE
      STATE
USE <PUTSTATE> WHEN STATE IS CORRECT.

TTASK ◇ PRINT ◇ NOHIGHSPEED
NOSTAMP ◇ NODIAMOND
WIDTH 132 ◇ DEPTH 66
ERASESRC ◇ OUTFILE 0
SET MARGINS TOP 2 BOTTOM 4 LEFT 0 RIGHT 0
KEYB NOMSG ◇ PAGEPAUSE 3 ◇ PAGENOS 1
PRINTSUMS ALPH ◇ ALPHASORT GRP, PKGS
PAGEFORMAT MULTIPLE, ONECOL FOR FNDEF, VARDEF
CENTRE TITLE BLANK
NONDEFAULT WSENV
LIST FNS, VARS, GRPS, GRP, FNSUM, VARSUM, GRPSUM, PKGSUM,
     FNDEF, VARDEF, MSGS
NOLIST BANNERS, COVER, TBLCONT, FNSUMC, PKGDEF, FNXREF,
       WSXREF, FNCALLS, WSGRAPH
NOFNDEF NONE
NOVARDEF NONE
SURROGATES 'BILN'

      PAGEPAUSE 1
      SAVESRC
      LIST WSXREF
      PUTSTATE 20
STATE REPLACED IN SRCFILE 20 - 1234567 UTILITY.
```

You're now ready to execute the *WSDOC* program.

Executing the *WSDOC* Program

Workspace 7 *STATE* contains the function *WSDOC*, which you can execute to load workspace 7 *WSDOC*. Alternately, you can load 7 *WSDOC* directly.

However 7 *WSDOC* is loaded, its profile is executed automatically. If you haven't defined a profile, the *WSDOC* program is begun immediately. (For information about how to set or reset a *WSDOC* profile, see "WSDOC Profiles".)

If your session parameter (system variable ⎕SP) is a numeric vector, the *WSDOC* program assumes that it contains WSDOC source file tie numbers, and begins processing the source files immediately. Otherwise, *WSDOC* prompts you for source file tie numbers.

How the *WSDOC* program proceeds is determined by the state stored in each source file. If you allow both the *TTASK* and *PRINT* options, the WSDOC report is produced on your terminal; you're prompted to *ALIGN PAPER*. Do so, and press RETURN; printing of the WSDOC report will begin immediately. (Both *TTASK* and *PRINT* are default settings.)

If you set *NTASK*, *BTASK*, or *NOPRINT*, and allow *LIST MSGS*, the *WSDOC* program displays information about the source and output files, and about any N-task started or B-task request submitted.

Examples. Suppose you want to document your workspace *UTILITY*.

- First, let's try it with the default state:

```
      )WSID
IS 7 STATE
      WSDOC
SRCFILE 20            - 1234567 UTILITY
WSID                  - 1234567 UTILITY
ALIGN PAPER
RETURN
[WSDOC report begins]
```

- Now let's try it as an N-task:

```
      WSDOC
SRCFILE 20            - 1234567 UTILITY
OUTFILE 40            - 1234567 UTILITY
WSID                  - 1234567 UTILITY
NTASK TASKID          - 7121 - CREATED
```

- Finally, let's try it as a B-task:

```
      WSDOC
SRCFILE 20            - 1234567 UTILITY
OUTFILE 40            - 1234567 UTILITY
WSID                  - 1234567 UTILITY
BTASK REQUEST         - 12513 - FILED
```

Documenting Several Workspaces at Once

You can use WSDOC easily to document several workspaces with only one pass through the *STATE* and *WSDOC* workspaces. This example should illustrate how:

```
      )LOAD A
SAVED  22.17.53 06/05/84
      )COPY 7 WSTOFILE
SAVED  20.30.31 03/18/84

      WSTOFILE
SRCFILE 20 - 1234567 A
11 FUNCTIONS
6 VARIABLES
1 GROUP

      )LOAD B
SAVED  23.18.54 07/06/84
      )COPY 7 WSTOFILE
SAVED  20.30.31 03/18/84

      WSTOFILE
SRCFILE 21 - 1234567 B
16 FUNCTIONS
9 VARIABLES
2 GROUPS
```

```
      )LOAD C
SAVED   0.19.55 08/07/84
      )COPY 7 WSTOFILE
SAVED  20.30.31 03/18/84

      WSTOFILE
SRCFILE 22 - 1234567 C
22 FUNCTIONS
12 VARIABLES
3 GROUPS

      )WSID
IS 7 STATE

      NTASK
      NOPRINT
      CREATEOUT
      PUTSTATE 20 21
STATE REPLACED IN SRCFILE 20 - 1234567 A.
STATE REPLACED IN SRCFILE 21 - 1234567 B.
      HIGHSPEED
      PUTSTATE 22
STATE REPLACED IN SRCFILE 22 - 1234567 C.
      WSDOC
SRCFILE 20          - 1234567 A
OUTFILE 40          - 1234567 A
WSID                - 1234567 A
SRCFILE 21          - 1234567 B
OUTFILE 40          - 1234567 A
WSID                - 1234567 B
SRCFILE 22          - 1234567 C
OUTFILE 40          - 1234567 A
WSID                - 1234567 C
HSPRINT REQUEST FOR - 1234567 A
PRINT SPECS: ERAS,REMO(BIGA)
DELIVERY INSTRUCTIONS: HOLD FOR PICKUP BY BILL CLARKE,
BANK OF MANHATTAN
RETURN
NTASK TASKID        - 7293 - CREATED
```

WSDOC produces three reports for workspaces *A*, *B*, and *C*—but puts these reports to a single output file—*A*. This file is printed at remote node *BIGA* (that is, New York City) by the *HSPRINT* facility.

Note that *CREATEOUT* is set for all three workspaces: this implies to WSDOC that the reports for all three workspaces are to be put to a single output file that WSDOC creates. (You can similarly use the *OUTFILE* option.) *HIGHSPEED* or *HSPRINT* is set only for the last workspace.

For more information about the *CREATEOUT*, *OUTFILE*, *HIGHSPEED*, and *HSPRINT* options, see "WSDOC File Options".

Chapter 3

THE WSDOC STATE

What Is the State?

The WSDOC state is the set of options that controls how WSDOC analyses and lists a source file. You display and change the state in the workspace 7 *STATE*; but the state that is used is stored in each source file.

Displaying the State

The **current state** is how the state options are currently set in the *STATE* workspace. You can display the current state with the *STATE* function. For example:

```
      )WSID
IS 7 STATE
      DEFAULT
      PAGEPAUSE 1
      SAVESRC
      WIDTH 120
      DIAMOND
      NO WSENV
      LIST WSXREF
      STATE
USE <PUTSTATE> WHEN STATE IS CORRECT.

TTASK ◇ PRINT ◇ NOHIGHSPEED
NOSTAMP ◇ DIAMOND ◇ NOQUOTES ◇ QUADFNS
WIDTH 120 ◇ DEPTH 66
SAVESRC ◇ OUTFILE 0
SET MARGINS TOP 2 BOTTOM 4 LEFT 0 RIGHT 0
KEYB NOMSG ◇ PAGEPAUSE 1 ◇ PAGENOS 1
PRINTSUMS ALPH ◇ ALPHASORT GRP, PKGS
PAGEFORMAT MULTIPLE, ONECOL FOR FNDEF, VARDEF
PAGEFORMAT MULTIPLE, TWOCOL FOR WSXREF
CENTRE TITLE BLANK
NO WSENV
LIST FNS, VARS, GRPS, GRP, FNSUM, VARSUM, GRPSUM, PKGSUM,
     FNDEF, VARDEF, WSXREF, MSGS
NOLIST BANNERS, COVER, TBLCONT, FNSUMC, PKGDEF, FNXREF,
       FNCALLS, WSGRAPH
NOFNDEF NONE
NOVARDEF NONE
SURROGATES 'BILN' ◇ XREFSYMBS '→⍎:⍞⎕'
```

The *STATE* program returns its output as its explicit result.

The State Options

The WSDOC state options fall into four classes:

Control options
File options
Format options
Content options

Control options specify the type of task that a WSDOC job is to run under, and parameters related to the task type. The control options are:

TTASK, *NTASK*, and *BTASK*
PRINT and *NOPRINT*
KEYB
PAGEPAUSE

File options control how WSDOC's source and output files are to be processed. The file options are:

ERASESRC and *SAVESRC*
OUTFILE and *CREATEOUT*
HIGHSPEED and *NOHIGHSPEED*
HSPRINT

Format options control what a WSDOC report looks like. The format options are:

WIDTH
DEPTH
SET MARGINS
TITLE
PAGENOS
TREEHEIGHT
PRINTSUMS
ALPHASORT
PAGEFORMAT
STAMP and *NOSTAMP*
DIAMOND and *NODIAMOND*

Content options control what is included in a WSDOC report. The content options are:

WSENV
LIST, *NOLIST*, *LISTONLY*, and *NOLISTONLY*
ONLYFNDEF and *NOFNDEF*
ONLYVARDEF and *NOVARDEF*
ONLYPKGDEF and *NOPKGDEF*
QUOTES and *NOQUOTES*
QUADFNS and *NOQUADFNS*
BRANCHFNS
TREEROOTS
SURROGATES
XREFSYMBS

You exercise most control over a WSDOC report's content with the *LIST*, *NOLIST*, *LISTONLY*, and

NOLISTONLY options. You specify to these options the **report elements** that you want included or excluded. The report elements are:

BANNERS	*VARSUM*	*WSGRAPH*
COVER	*GRPSUM*	*MSGS*
TBLCONT	*PKGSUM*	*SYSCMDS*
FNS	*FNDEF*	*SUMS*
VARS	*VARDEF*	*DEFS*
GRPS	*PKGDEF*	*XREFS*
GRP	*FNXREF*	*ALL*
FNSUM	*WSXREF*	*NONE*
FNSUMC	*FNCALLS*	

The Default State

If a WSDOC source file has no state stored in it, the *WSDOC* program uses the **default state** when it processes that file. When a WSDOC source file is created, no state is stored in it; consequently, the default state applies to a source file unless you put the state from the *STATE* workspace to the file.

The default state is in effect when you load the *STATE* workspace:

- unless your *STATE* profile—if you have one—changes the state; or
- until you change the state yourself in immediate-execution mode.

WSDOC produces quite useful reports when the default state is in effect. The default state assumes that you want:

- the report printed at your terminal;
- all functions, variables, groups, and packages summarized;
- functions and variables defined; and
- no cross-reference tables.

You can restore the default state at any time by executing *DEFAULT*.

Changing the State

You change the state by executing **state-setting functions**. For example, to reset the *DIAMOND/NODIAMOND* state, you execute the *DIAMOND* or *NODIAMOND* function.

Putting the State to File

The state that WSDOC uses is the state that is stored in each source file. Thus, when the current state in the *STATE* workspace is how you want it, you have to store it in the appropriate source files.

This is done with the *PUTSTATE* program:

```
      PUTSTATE 21 22
STATE REPLACED IN SRCFILE 21 - 1234567 BETA.
STATE REPLACED IN SRCFILE 22 - 1234567 GAMMA.
```

```
      PUTSTATE 'DELTA'
STATE REPLACED IN SRCFILE 23 - 1234567 DELTA.
```

PUTSTATE takes as its right argument the name of a single WSDOC source file, or a scalar or vector of source file tie numbers.

Getting the State from File

You can copy the state from a WSDOC source file into the *STATE* workspace with the *GETSTATE* program. *GETSTATE* takes as its right argument the name or tie number of a single WSDOC source file.

You would typically use *GETSTATE* when you wanted to modify a state already stored. For example:

```
      LIST FNXREF
      PUTSTATE 'DELTA'
STATE REPLACED IN SRCFILE 23 - 1234567 DELTA.
      ...
      GETSTATE 'DELTA'
STATE RETRACTED FROM SRCFILE 23 - 1234567 DELTA.
      NOLIST FNXREF
      LIST WSXREF
      PUTSTATE 23
STATE REPLACED IN SRCFILE 23 - 1234567 DELTA.
```

You might also use *GETSTATE* to copy a state from one source file to another. For example:

```
      GETSTATE 'EPSILON'
STATE RETRACTED FROM SRCFILE 20 - 1234567 EPSILON.
      PUTSTATE 'ZETA'
STATE REPLACED IN SRCFILE 21 - 1234567 ZETA.
```

Chapter 4

WSDOC CONTROL OPTIONS

The WSDOC state control options are used to specify the type of task that a WSDOC job is to run under, and to specify parameters related to the task type. The control options are:

TTASK, *NTASK*, and *BTASK*
PRINT and *NOPRINT*
KEYB
PAGEPAUSE

Syntax: Square brackets—[ ]—indicate that an element is optional. Braces and staves indicate that you must choose only one element from a list; for example, {*TTASK*|*NTASK*|*BTASK*} indicates that you must enter *TTASK* or *NTASK* or *BTASK*.

The *TTASK*, *NTASK*, and *BTASK* Options

{*TTASK*|*NTASK*|*BTASK*}
Specifies the type of SHARP APL task that the WSDOC job is to run under.

Default: *TTASK*

Usage: You can run a WSDOC job:

- in your current terminal session (that is, in your current T-task);
- in an N-task; or
- in a B-task.

By default, WSDOC jobs are run in your current T-task.

For more information about N- and B-tasks as such, see the manual entitled *Batch Tasks in SHARP APL.*

When you execute the *WSDOC* program on several source files, WSDOC sorts the files by task type. One N-task is started for all N-task WSDOC jobs (if any), one B-task request is submitted for all B-task WSDOC jobs, and then the T-task WSDOC jobs are begun.

T-task: Running WSDOC in your current T-task is the fastest but also the most expensive way to produce a WSDOC report. The report is printed (or appended to the output file) as it is composed.

While your T-task WSDOCs are running, your terminal won't accept input. You can press ATTN or BREAK to interrupt a WSDOC. If you want to resume the WSDOC, type:

```
→GOON
```

N-tasks: Running WSDOC in an N-task is nearly as fast, but not as expensive as running it in a T-task.

An N-task begins executing right away; WSDOC displays the N-task's task number. N-tasks are charged for system resources at rates lower than those for T-tasks and higher than those for B-tasks.

If an N-task WSDOC is abnormally terminated, the *WSDOC* workspace at the moment of termination is saved in your workspace library with the name *WSDOCC*. An N-task WSDOC job is abnormally terminated if the *WSDOC* program encounters an error it can't recover from (such as *WS FULL*), or if the N-task is bounced.

You can restart an interrupted WSDOC by loading *WSDOCC* and typing *→GOON*. The WSDOC job resumes in your T-task.

B-tasks: Running WSDOC in a B-task is much less expensive than running it in a T- or N-task.

WSDOC submits a B-task request right away, but the B-task won't be started until system load is low. B-tasks are charged for system resources at rates much lower than those for T- and N-tasks.

Workspace 1 *BTASKREQ* contains functions you can use to withdraw a B-task request, and to display information about a request. If you withdraw a WSDOC B-task request, the WSDOC is not run. However, you can use the *WSDOC* program to resubmit the request.

If a B-task WSDOC is abnormally terminated, the *WSDOC* workspace at the moment of termination is saved in your workspace library with the name *WSDOCC*. As with an N-task, a B-task WSDOC is abnormally terminated if the *WSDOC* program encounters an error it can't recover from (such as *WS FULL*), or if the B-task is bounced.

You can restart an interrupted WSDOC by loading *WSDOCC* and typing *→GOON*. The WSDOC job resumes in your T-task.

The *PRINT* and *NOPRINT* Options

{PRINT|NOPRINT}
Specifies whether the WSDOC report is to be printed at your terminal while it is being composed.

Default: *PRINT*

Usage: If you set *NOPRINT*, you must also set *OUTFILE* or *CREATEOUT*; if you don't, *WSDOC* prints an error message, and aborts analysis of the offending source file.

The *KEYB* Option

KEYB {NOMSG|NORMAL}
Specifies whether your terminal can receive messages while the WSDOC report is being composed. *NOMSG* indicates that your terminal can't receive messages; *NORMAL* indicates that your terminal can.

Default: *KEYB NOMSG*

Usage: This option is ignored if *NOPRINT* is set.

The *PAGEPAUSE* Option

PAGEPAUSE **n**
Specifies that printing of the WSDOC report is to pause every **n** pages. To restart printing, you must press RETURN. You can use this interval to re-adjust paper.

Default: *PAGEPAUSE* 3

Minimum: *PAGEPAUSE* 1 (*PAGEPAUSE* 0 suppresses pausing)

Maximum: *PAGEPAUSE* 1000

Usage: This option is ignored if *NOPRINT* is set.

WSDOC FILE OPTIONS

The WSDOC state file options are used to control how WSDOC's source and output files are to be processed. The file options are:

ERASESRC and *SAVESRC*
OUTFILE and *CREATEOUT*
HIGHSPEED and *NOHIGHSPEED*
HSPRINT

Syntax: Square brackets—[ ]—indicate that an element is optional. Braces and staves indicate that you must choose only one element from a list; for example, {*ERASESRC*|*SAVESRC*} indicates that you must enter either *ERASESRC* or *SAVESRC*.

The *ERASESRC* and *SAVESRC* Options

{*ERASESRC*|*SAVESRC*}
Specifies whether the source file is to be erased when the WSDOC report is done. The source file is the file created by the *WSTOFILE* program.

Default: *ERASESRC*

Usage: One reason for selecting *SAVESRC* is if you intend to run another WSDOC of the same workspace with different state settings in effect. You might also wish to retain the source file and update the objects in it using the WSDOC utility programs.

The *OUTFILE* and *CREATEOUT* Options

OUTFILE {0|**n**}|*CREATEOUT*
Specifies an **output file**; that is, a SHARP APL file to which the WSDOC report is to be appended. *OUTFILE* 0 indicates that the WSDOC report is not to be appended to a file. *OUTFILE* **n** indicates that the WSDOC report is to be appended to the file tied to **n**. *CREATEOUT* indicates that WSDOC is to create a file to which it is to append its report.

Default: *OUTFILE* 0

Usage: The report that WSDOC appends to the output file is identical whether you specify *OUTFILE* or *CREATEOUT*. Whichever you use, the report includes *HSPRINT* control messages so that it can be produced through the *HSPRINT* facility.

WSDOC appends its report to the output file. This means that if a file you specify with the *OUTFILE* option already has data, this data is not disturbed: the WSDOC report comes after the old data.

If you set *CREATEOUT*, WSDOC tries to create the output file when you execute the *WSDOC* program. The name of the created file is the same as the name of the workspace being documented, but with the first character of the name underscored; for example, if you're documenting a workspace named *UTILITY*, WSDOC tries to create an output file named *_UTILITY*. Conversely, if the work-

space name's first character is already underscored, the output file name's first character is not underscored; for example, if you're documenting workspace *UTILITY*, WSDOC tries to create an output file named *UTILITY*.

If a file with the derived name already exists in your library, the *WSDOC* program prompts you to specify an alternate file name, or to exit from the program by entering SPACE RETURN.

WSDOC ties a created output file to the lowest available tie number greater than or equal to 40.

You needn't tie the file specified by *OUTFILE* **n** until just before you execute the *WSDOC* program.

If you set or allow *PRINT*, you may also set *OUTFILE* **n** or *CREATEOUT*; that is, you can have a WSDOC report both printed on your terminal as it is being composed, and simultaneously appended to an output file.

The *HIGHSPEED* and *NOHIGHSPEED* Options

{HIGHSPEED|NOHIGHSPEED}
Specifies whether WSDOC is to prepare and submit an *HSPRINT* request for the output file when the report is done.

Default: *NOHIGHSPEED*

Usage: If you set *HIGHSPEED*, you must also set *OUTFILE* or *CREATEOUT*.

If you set *HIGHSPEED* and allow *TTASK* (that is, if the WSDOC report is composed in your current terminal session), you're prompted for *HSPRINT* print specifications and delivery instructions after the report is composed, and the *HSPRINT* submission or error message is displayed on your terminal.

If you set *HIGHSPEED* and *NTASK* or *BTASK*, you're prompted for *HSPRINT* print specifications and delivery instructions before WSDOC starts the N-task or submits the B-task request.

The *HSPRINT* Option

HSPRINT
Changes *PRINT* to *NOPRINT*, *NOHIGHSPEED* to *HIGHSPEED*, and *OUTFILE* 0 to *CREATEOUT*. Options not at their default settings are not changed.

Usage: *HSPRINT* is the only state-setting function that normally has output: *HSPRINT* displays the options it has reset. From the default state, *HSPRINT* displays "*NOPRINT* ◇ *HIGHSPEED* ◇ *CREATEOUT*". If you've already set *HIGHSPEED*, *HSPRINT* displays "*NOPRINT* ◇ *CREATEOUT*".

If you've set *NOPRINT*, *HIGHSPEED*, and *CREATEOUT* or *OUTFILE* already, *HSPRINT* displays nothing and has no effect.

If you set *OUTFILE* **n** (where **n** is not zero), *HSPRINT* does not set *CREATEOUT*.

Example. The user starts an N-task WSDOC job that will end in an *HSPRINT* request:

```
        NTASK
        HSPRINT
NOPRINT ◇ HIGHSPEED ◇ CREATEOUT
```

```
        PUTSTATE 20
STATE REPLACED IN SRCFILE 20 - 1234567 UTILITY.
        WSDOC
SRCFILE 20          - 1234567 UTILITY
OUTFILE 40          - 1234567 UTILITY
WSID                - 1234567 UTILITY
HSPRINT REQUEST FOR - 1234567 UTILITY
PRINT SPECS: ERAS,REMO(PALO)
DELIVERY INSTRUCTIONS: HOLD FOR PICKUP BY MELLOW COURIERS INC.
RETURN
NTASK TASKID        - 7188 - CREATED
```

Chapter 6

WSDOC FORMAT OPTIONS

The WSDOC state format options are used to control what a WSDOC report looks like; that is, its page format. The format options are:

WIDTH
DEPTH
SET MARGINS
TITLE
PAGENOS
TREEHEIGHT
PRINTSUMS
ALPHASORT
PAGEFORMAT
STAMP and *NOSTAMP*
DIAMOND and *NODIAMOND*

Syntax: Square brackets—[ ]—indicate that an element is optional. Braces and staves indicate that you must choose only one element from a list; for example, *PRINTSUMS {ALPH|DESCENDING|ASCENDING}* indicates that you must enter *PRINTSUMS ALPH* or *PRINTSUMS DESCENDING* or *PRINTSUMS ASCENDING*.

Page Format

WSDOC report page format is illustrated on the next page. Every page of a WSDOC report has **margins (top** and **bottom, left** and **right)**. Every page except the banners, the title page, and the table of contents has a **title** and a **header**. What is left over is the **report area**.

A page's dimensions—**width** and **depth**—are set by the *WIDTH* and *DEPTH* options. The **effective width** is what's left over after you subtract the left and right margins. The **effective depth** is what's left over after you subtract the top and bottom margins.

Margins are controlled with the *SET MARGINS* option, and the title with the *TITLE* option. The *PAGENOS* option is used to control the header's *PAGE* field. The left-hand field of the header (the workspace name and save time) cannot be suppressed.

TITLE

```
APL CHESS PLAYING PROGRAMS
```

WORKSPACE IDENTIFICATION

TIMESTAMP LAST SAVED

```
1234567 CHESS    *    SAVED  17.54.28 06/12/84
```

PAGE NUMBER

```
PAGE   25
```

FIRST AND LAST FUNCTIONS ON PAGE

```
FNS: PCAPS - QI
```

FUNCTION DEFINITION

```
    ∇ Z←PCAPS;TO;MASK
[1]   ⍝ COMPUTE PAWN CAPTURES.
[2]     TO←,PCINC[COLOUR;]∘.+SQ ⋄ MASK←TO∊OPPSQUARES
[3]     Z←(MASK/(2×⍴SQ)⍴SQ),[1.5] MASK/TO
    ∇
```

FUNCTION CROSS-REFERENCE

```
COLOUR          *   2                          TO          LV   0   2←  2   3
MASK            LV  0   2←  3   3              Z           RS   0←  3←
OPPSQUARES      *   2                          PCINC       GV   2[
SQ              *   2   3   3
```

```
    ∇ Z←PMOVE COLOUR;A;B;C;TA;TB;TC;FLG;FROM;MASK
[1]   ⍝ COMPUTE POSSIBLE PAWN MOVES FOR THE SPECIFIED COLOUR.
[2]     FLG← 1*COLOUR=BLACK
[3]     FROM←(4×⍴SQ)⍴SQ ⋄ TA←SQ+SMPINC×FLG ⋄ A←CC>TA∊OCCSQUARES
[4]     TB←,(CMPINC×FLG)∘.+SQ ⋄ B←CC≤TB∊OPPSQUARES
[5]     TC←SQ+DMPINC×FLG ⋄ C←A∧(SQ∊(8×-FLG)↑(8×-FLG)↓BOARD)>TC∊OCCSQUARES
[6]     MASK←A,B,C ⋄ Z←(MASK/FROM),[1.5] MASK/TA,TB,TC
    ∇

A               LV  0   3←  5   6              SQ          *   3   3   3   4   5   5
B               LV  0   4←  6                  TA          LV  0   3←  3   6
C               LV  0   5←  6                  TB          LV  0   4←  4   6
CC              *   3   4                      TC          LV  0   5←  5   6
COLOUR          RA  0   2                      Z           RS  0←  6←
FLG             LV  0   2←  3   4   5   5   5  BLACK       GV  2
FROM            LV  0   3←  6                  BOARD       GV  5
MASK            LV  0   6←  6   6              CMPINC      GV  4
OCCSQUARES      *   3   5                      DMPINC      GV  5
OPPSQUARES      *   4                          SMPINC      GV  3
```

```
    ∇ Z←FLG QI INP
[1]   ⍝ PROMPT THE USER FOR KEYBOARD INPUT.
[2]     ⍎(0=⎕NC 'FLG')/'FLG←0' ⍝ ASSUME REPROMPT ON NULL RESPONSE
[3]   L0:⍞←INP ⋄ ⎕ARBOUT '' ⋄ →L0×FLG<0∊⍴Z←⍞
    ∇

⎕ARBOUT         QF  3                          Z           RS  0←  3←
⎕NC             QF  2                          →           *   3
FLG             LA  0   2   2←  3              ⍎           *   2
INP             RA  0   3                      :           *   3
L0              LL  3:  3                      ⍞           *   3←  3
```

The *WIDTH* Option

WIDTH **n**
Specifies that the WSDOC report's pages are to be **n** characters wide.

Default: *WIDTH* 132

Minimum: *WIDTH* 30

Maximum: *WIDTH* 250

Usage: The *WIDTH* setting includes the columns set aside for the left and right margins. The effective width—the number of columns available for the report itself—must be at least 30 characters.

The *DEPTH* Option

DEPTH **n**
Specifies that the WSDOC report's pages are to be **n** lines deep.

Default: *DEPTH* 66

Minimum: *DEPTH* 20

Usage: The *DEPTH* setting includes the lines set aside for the top and bottom margins. The effective depth—the page area available for the report itself—must be at least 20 lines; consequently, if you allow the default top and bottom margins (which total 6 lines), *DEPTH* must be set to at least 26.

The *SET MARGINS* Option

SET MARGINS [*TOP* **j**] [*BOTTOM* **k**] [*LEFT* **m**] [*RIGHT* **n**]
Specifies that the top and bottom margins are to be **j** and **k** lines deep, respectively; and that the left and right margins are to be **m** and **n** characters wide, respectively.

Default: *SET MARGINS TOP* 2 *BOTTOM* 4 *LEFT* 0 *RIGHT* 0

Usage: You can set one or several margins, in any order, in a single statement; for example:

```
SET MARGINS LEFT 15
SET MARGINS RIGHT 10 LEFT 15 TOP 6
```

The *TITLE* Option

[*CENTRE*|*CENTER*] *TITLE* {*BLANK*|**'string'**}
Specifies a title that is to appear on every page of the WSDOC report. *CENTRE* or *CENTER* indicates that the title is to be centered laterally between the left and right margins. *BLANK* specifies that the title is to be blank. **string** is the title text; it may be of any length, and may include embedded carriage returns.

Default: *CENTRE TITLE BLANK*

Examples. A simple title:

```
      TITLE 'DAVID FALKAYN''S UTILITY WORKSPACE'
```

A title over several lines:

```
      T←'A GENERALIZED FINANCIAL PLANNING SYSTEM',2ρCR←⎕AV[⎕IO+156]
      T←T,'WRITTEN BY ALBERT CORMIER',CR
      T←T,'BANQUE DU NOUVEAU MONDE / '
      ⎕←T←T,'SERVICE D''INFORMATIQUE',2ρCR
A GENERALIZED FINANCIAL PLANNING SYSTEM

WRITTEN BY ALBERT CORMIER
BANQUE DU NOUVEAU MONDE / SERVICE D'INFORMATIQUE


      CENTRE TITLE T
```

A title with two fields, one to be positioned at the far left, the other at the far right:

```
      TITLE T,(ρT←'LEFT END')↓¯132↑'RIGHT END'
```

The *PAGENOS* Option

PAGENOS **n**
Specifies either the initial page number, or all page numbers.

If **n** is a single integer (scalar or one-element vector), **n** specifies the initial page number; all subsequent page numbers are incremented by 1 from **n**.

If **n** is a vector of several integers, each element of the vector is used in turn as a page number. If there are fewer elements than pages, the last pages are unnumbered.

Default: *PAGENOS* 1

Usage: Banners, title pages, and tables of contents aren't given page numbers.

You can suppress page numbers by setting *PAGENOS* ⍳0. If you do this, the word "*PAGE*" does not appear in the header.

PAGENOS' argument, **n**, can include negative numbers, zero, and real numbers (which are rounded to integers). However, WSDOC can display page numbers only in the range ¯999 to 9999, inclusive; numbers beyond this range are shown as ****.

You would likely specify a vector of page numbers when you want to update only certain pages of a WSDOC report. For example, if you produced the original report with *PAGEFORMAT SINGLE*, *ONECOL FOR FNDEFS* set, you can update the report only for changed functions by careful use of *PAGENOS* and *ONLYFNDEF*.

Examples. This starts page numbering with 10 (rather than 1):

```
      PAGENOS 10
```

The following statement sets arbitrary page numbers for the report's first ten pages; thereafter, pages are unnumbered:

```
PAGENOS 15 17 33 43 43 44 72 75 98 100
```

To suppress page numbers and the word "*PAGE*", use:

```
PAGENOS ⍳0
```

The *TREEHEIGHT* Option

TREEHEIGHT **n**

Specifies that **n** is the maximum number of lines which a subtree may occupy in the tree graph structure diagram. **n** may be any non-negative integer. If **n** is 0, subtrees are not limited.

Default: *TREEHEIGHT* 10

Usage: If a called function's subtree will exceed **n** lines, it is broken out as a branch function with its own tree. Branch functions are marked with ⊛ after their names. The continuation of the branch appears as a root elsewhere in the tree graph output, but with the name of the branch preceded by ⊛.

WSDOC attempts to keep any workspace structure tree on a single page; this also works to limit the size of trees and subtrees.

For more information about the tree graph structure diagram, see "The *BRANCHFNS* Option" and "The *WSGRAPH* Option".

If you set *TREEHEIGHT* 0, your WSDOC job may fail on a *WS FULL* error.

The *PRINTSUMS* Option

PRINTSUMS {*ALPH*|*ASCENDING*|*DESCENDING*}

Specifies the order in which the identifiers in the function, variable, group, and package summaries are to be selected for printing. *ALPH* specifies that the identifiers are to be selected in alphabetical order. *ASCENDING* specifies that the identifiers are to be selected in ascending order according to the space occupied by each identified object (that is, smallest first). *DESCENDING* specifies that the identifiers are to be selected in descending order according to the space occupied by each identified object (that is, largest first).

Default: *PRINTSUMS ALPH*

The *ALPHASORT* Option

ALPHASORT {[*FNS*][,*VARS*][,*GRPS*][,*GRP*][,*PKGS*]|*ALL*|*NONE*}

Specifies which classes of identifiers in the WSDOC source file are to be sorted alphabetically before the file is analysed. *FNS* specifies that functions are to be sorted; *VARS* specifies that variables are to be sorted; *GRPS* specifies that groups are to be sorted; *GRP* specifies that the members of each group are to be sorted; *PKGS* specifies that packages and the identifiers within each package are to be sorted at every level; and *ALL* specifies that all identifiers are to be sorted. *NONE* indicates that no class of identifiers is to be sorted.

Default: *ALPHASORT GRP*, *PKGS*

Usage: SHARP APL maintains functions, variables, and groups in alphabetically sorted order in each workspace. Consequently, if you're using WSDOC to analyse a source file generated from a single workspace, you need set or allow only *ALPHASORT GRP*, *PKGS* (the default) to make sure that all identifiers are sorted.

If you're using WSDOC to analyse a source file generated from several workspaces or program files, you can set *ALPHASORT* to make sure that identifiers are sorted. (This would occur if you used the *GENWSTOFILE* or *MRGWSTOFILES* program to construct the source file.)

WSDOC lists and defines objects in the order in which they're found in the source file, unless you set *ALPHASORT*. Thus, *ALPHASORT* potentially affects the output selected by *LIST* elements *FNS*, *VARS*, *GRPS*, *GRP*, *FNDEF*, *VARDEF*, *PKGDEF*, *FNCALLS*, and *WSGRAPH*.

Identifiers in cross-reference tables are always sorted alphabetically. Identifiers in object summaries are always sorted according to the *PRINTSUMS* option.

The *PAGEFORMAT* Option

PAGEFORMAT [*SINGLE*|*MULTIPLE*|*RUNNING*] [,*ONECOL*|*TWOCOL*]
[*FOR* [*FNDEF*|,*VARDEF*|,*PKGDEF*|,*FNXREF*|,*WSXREF*|,*DEFS*|,*XREFS*]]

Controls how object definitions and cross-reference tables are to be grouped and formatted on the WSDOC report's pages. (Object definitions comprise function, variable, and package definitions.)

SINGLE specifies that only one object definition or cross-reference table is to be printed in each column. *MULTIPLE* specifies that as many object definitions or function cross-reference tables as will fit completely are to be printed in each column; and that definitions are not to be run on to the next column unless the whole definition by itself takes more than one column. *RUNNING* specifies that definitions are to be printed continuously, without concern for where column or page breaks appear relative to where objects begin.

ONECOL specifies that object definitions or cross-reference tables are to be formatted in a single column across the page. *TWOCOL* specifies that object definitions or cross-reference tables are to be formatted in two columns on each page; each column is slightly less than half the *WIDTH* setting wide.

FOR specifies that the format specifications apply to the following report elements. If you omit this operand, the *PAGEFORMAT* option applies to all selected report elements. *FNDEF* specifies that the format specifications apply to function definitions, *VARDEF* to variable definitions, *PKGDEF* to package definitions, *FNXREF* to the function cross-reference tables, and *WSXREF* to the workspace cross-reference table. *DEFS* specifies that the format specifications apply to all object definitions. *XREFS* specifies that the format specifications apply to all cross-reference tables.

Default: *PAGEFORMAT MULTIPLE*, *ONECOL FOR DEFS* ◇ *PAGEFORMAT MULTIPLE*, *TWOCOL FOR XREFS*

Usage: If you set or allow *PAGEFORMAT SINGLE* or *MULTIPLE*, an object definition or cross-reference table will run over one column if, even by itself, it can't fit in a single column.

If you allow or set *PAGEFORMAT TWOCOL FOR XREFS*, the cross-reference tables may nonetheless be formatted in a single column if too few references would fit on a line. This can happen if the length of the longest identifier is substantial compared with the page width.

If you set or allow *PAGEFORMAT MULTIPLE FOR FNDEF*, *FNXREF*, and *LIST FNDEF*, *FNXREF*, a function definition will be started on a page only if the function's cross-reference table will also fit completely. This format is especially propitious because it locates a function and its cross-reference table on the same page wherever possible.

If you want to set different format specifications for different groups, you must use *PAGEFORMAT* repeatedly for each format specification. For example:

```
PAGEFORMAT TWOCOL FOR FNDEF
PAGEFORMAT RUNNING, TWOCOL FOR VARDEF
PAGEFORMAT ONECOL FOR WSXREF
```

If you set *PAGEFORMAT TWOCOL* for any object definitions, those definitions will be formatted in two columns whatever the setting of *WIDTH*. However, you're likely to find your function definitions hard to read unless *WIDTH* is at least 100, or your function lines are unusually short.

The *STAMP* and *NOSTAMP* Options

{STAMP|NOSTAMP}
Specifies whether a timestamp is to be displayed with each object definition. This timestamp shows when an object's image was appended to or replaced in the source file.

Default: *NOSTAMP*

Usage: If you allow *NOSTAMP*, object definitions are separated by a single blank line. If you set *STAMP*, definitions are separated by three lines, the first and third blank, and the second containing a timestamp like this at the left margin:

```
22.27.18  11 AUG 1984
```

The *DIAMOND* and *NODIAMOND* Options

{DIAMOND|NODIAMOND}
Specifies whether function lines in function definitions are to be reformatted. If you select *DIAMOND*, function lines that contain two or more statements (separated by ◇ 'diamond') are shown with each statement starting on a new line; and the start of each statement (after any label) is indented past the longest label, so that all statements start in the same column.

Default: *NODIAMOND*

Example. This is a function definition with *DIAMOND* set:

```
    ∇ Z←DMATE COLOUR;T;KSQ;MOVES;PINNINGSQ
[1]       ⍝ DETERMINE IF PLAYER IS IN A MATE SITUATION.
[2]        Z←0 ◇
           MOVES←LEGAL∆MOVES COLOUR
[3]   L0: →('INCHECK COLOUR' TEST∆MOVE 2⍴MOVES)↓0
[4]        KSQ←POSITION[1;POSITION[2;]⍳COLOUR SETC KING]
[5]       →(T[1]=KSQ)⍴L1
[6]        PINNINGSQ←(KSQ=SUPPORTMOVES[;2])/SUPPORTMOVES[;1]
[7]       →(1∊MOVES∧.=T[1],[0.5] PINNINGSQ)⍴0
[8]        MOVES←(T[1]≠MOVES[;1])⌿MOVES ◇
          →L0×⍴MOVES
[9]   L1: →L0×~Z←0∊⍴MOVES← 1 0 ↓MOVES
    ∇
```

Alternate Definition Formats

When you execute *WSTOFILE* it sets up function, variable, and group index components in the source file, in components 17, 20, and 23, respectively. These index components are integer vectors indicating where the source file contains the definition of each identifier. *WSDOC* uses the definition of each object for performing calculations and cross-referencing, as well as for displaying the identifiers in the *FNDEF*, *VARDEF*, and *PKGDEF* options. Normally, each object is displayed the same way it is at a terminal, honouring the workspace environment settings (⎕*PP* and ⎕*PS*) in the source workspace.

You can change the format of display used by *WSDOC* for these options as follows. First, append the objects in the format you desire to the source file. The objects may be character or numeric, and may be of any rank and shape. Then, the index directories for the class or classes of objects just appended must be augmented, by making them two-column matrices rather than vectors. The first column of the matrix contains the original index directory, and the new second column, the directory pointers to the alternately formatted objects in the file. If an element in the second column of a directory is 0, then the default format for the identifier corresponding to that row will be used. In this manner, you can specify a subset of functions or variables to be printed in a different format, leaving all others in the standard format.

The ability to use a different format for displaying an object in a WSDOC report is particularly useful for application packages that use WSDOC for documentation, but that maintain their own independent function and variable format. Alternate formatting is also useful if you want to place any additional information or commentary on the front of each program or variable that is displayed.

Because WSDOC requires a standard-format definition of each identifier for processing, you should not change definitions of the objects set up by *WSTOFILE*, nor alter the first column of the pointer directories.

Augmenting the group index in the source file is permitted; however, the additional column is ignored.

Chapter 7

WSDOC CONTENT OPTIONS

The WSDOC state content options are used to control what is included in a WSDOC report. The content options are:

WSENV
LIST, *NOLIST*, *LISTONLY*, and *NOLISTONLY*
ONLYFNDEF and *NOFNDEF*
ONLYVARDEF and *NOVARDEF*
ONLYPKGDEF and *NOPKGDEF*
QUOTES and *NOQUOTES*
QUADFNS and *NOQUADFNS*
BRANCHFNS
TREEROOTS
SURROGATES
XREFSYMBS

Syntax: Square brackets—[ ]—indicate that an element is optional. Braces and staves indicate that you must choose only one element from a list; for example, *{NO|NONDEFAULT|FULL} WSENV* indicates that you must enter *NO WSENV* or *NONDEFAULT WSENV* or *FULL WSENV*.

The *WSENV* Option

{NO|NONDEFAULT|FULL} WSENV
Specifies the type of workspace environment display that is to appear in the WSDOC report. *NO* indicates that the workspace environment display is not to be produced. *NONDEFAULT* indicates that the display is to include only those parameters whose values differ from their default values in a clear workspace. *FULL* indicates that the full display is to be produced.

Default: *NONDEFAULT WSENV*

Usage: A full workspace environment display shows:

- workspace owner;
- environment condition (⎕*EC*);
- index origin (⎕*IO*);
- print precision (⎕*PP*);
- positioning and spacing (⎕*PS*);
- print width (⎕*PW*);
- comparison tolerance (⎕*CT*);
- random link (⎕*RL*);
- workspace available (⎕*WA*);
- percentage workspace used;
- number of symbols reserved;
- number of symbols used;
- latent expression (⎕*LX*);
- global trap definition (⎕*TRAP*);
- event report (⎕*ER*); and
- state indicator stack.

Below is an example of a workspace environment display produced with the default setting *NONDEFAULT WSENV*.

```
APL CHESS PLAYING PROGRAMS
1234567 CHESS * SAVED  17.54.28 06/12/84                                    PAGE 1

NONDEFAULT WORKSPACE ENVIRONMENT

⎕PP          - 18
⎕PS          - ¯1 ¯1 ¯2 ¯3
⎕WA          - 266540
⍟/⍟ WS USED - 11
SYMB RSVD    - 545
SYMB USED    - 257
⎕LX          - CHESS
```

The *LIST*, *NOLIST*, *LISTONLY*, and *NOLISTONLY* Options

LIST **element1, element2, ...**
Specifies that **element1** etc. are to be included in the WSDOC report.

NOLIST **element1, element2, ...**
Specifies that **element1** etc. are to be excluded from the WSDOC report.

LISTONLY **element1, element2, ...**
Specifies that **element1** etc. are to be the only elements included in the WSDOC report.

NOLISTONLY **element1, element2, ...**
Specifies that **element1** etc. are to be the only elements excluded from the WSDOC report.

element1, element2, ...
Are selectable WSDOC report elements, specifically:

BANNERS	*VARSUM*	*WSGRAPH*
COVER	*GRPSUM*	*MSGS*
TBLCONT	*PKGSUM*	*SYSCMDS*
FNS	*FNDEF*	*SUMS*
VARS	*VARDEF*	*DEFS*
GRPS	*PKGDEF*	*XREFS*
GRP	*FNXREF*	*ALL*
FNSUM	*WSXREF*	*NONE*
FNSUMC	*FNCALLS*	

Default: *LIST FNS, VARS, GRPS, GRP, FNSUM, VARSUM, GRPSUM, PKGSUM, FNDEF, VARDEF, MSGS* ◇ *NOLIST BANNERS, COVER, TBLCONT, FNSUMC, PKGDEF, FNXREF, WSXREF, FNCALLS, WSGRAPH*

Usage: Every time you select the *LIST* or *NOLIST* option, you affect only those report elements you specify in the option. Each *LIST* or *NOLIST* updates the selection of report elements. For example:

```
DEFAULT
LIST FNSUMC, FNXREF, WSGRAPH
NOLIST GRPSUM, VARDEF
LIST VARDEF, PKGDEF
```

changes the default selection by including *FNSUMC*, *PKGDEF*, *FNXREF*, and *WSGRAPH*, and by excluding *GRPSUM*. *VARDEF* is excluded and then included again.

Every time you select the *LISTONLY* or *NOLISTONLY* option, you completely alter the selection of report elements; every element is affected, whether specified or not. For example:

```
LISTONLY FNSUMC, FNDEF, WSGRAPH, MSGS
```

alters the current report element selection by excluding every element but *FNSUMC*, *FNDEF*, *WSGRAPH*, and *MSGS*.

LISTONLY ... is equivalent to *NOLIST ALL* ◇ *LIST* ....

NOLISTONLY ... is equivalent to *LIST ALL* ◇ *NOLIST* ....

The *ONLYFNDEF* and *NOFNDEF* Options

ONLYFNDEF **'names'**|*NOFNDEF* {*NONE*|['⍒'][,][**'names'**]}

Controls which functions are to be defined in the WSDOC report. *ONLYFNDEF* specifies functions that are to be the only functions defined in the report. *NOFNDEF* specifies functions that are not to be defined in the report. *NOFNDEF NONE* indicates that all functions are to be defined. ⍒ indicates that locked functions are not to be listed with other functions' definitions. **names** is a list of function or group names separated by spaces or commas.

Default: *NOFNDEF NONE*

Usage: *ONLYFNDEF* and *NOFNDEF* are mutually exclusive. However, if you select *ONLYFNDEF*, *NOFNDEF* '⍒' is implied: locked functions in **names** are not listed.

The identifiers named in **names** need not exist in the source file, and need not identify valid functions or groups.

Each group named in **names** is expanded into its composite members. Thus, specifying a group to *ONLYFNDEF* or *NOFNDEF* has the same effect as specifying each of the group's members.

You can use *ONLYFNDEF* with *PAGENOS* and *PAGEFORMAT SINGLE FOR FNDEF* to update only certain pages of a WSDOC report.

The *ONLYVARDEF* and *NOVARDEF* Options

ONLYVARDEF **'names'**|*NOVARDEF* {*NONE*|**'names'**}

Controls which variables are to be defined in the WSDOC report. *ONLYVARDEF* specifies variables that are to be the only variables defined in the report. *NOVARDEF* specifies variables that are not to be defined in the report. *NOVARDEF NONE* indicates that all variables are to be defined. **names** is a list of variable or group names separated by spaces or commas.

Default: *NOVARDEF NONE*

Usage: The identifiers named in **names** need not exist in the source file, and need not identify valid variables or groups.

Each group named in **names** is expanded into its composite members. Thus, specifying a group to *ONLYVARDEF* or *NOVARDEF* has the same effect as specifying each of the group's members.

You can use *ONLYVARDEF* with *PAGENOS* and *PAGEFORMAT SINGLE FOR VARDEF* to update only certain pages of a WSDOC report.

The *ONLYPKGDEF* and *NOPKGDEF* Options

ONLYPKGDEF **'names'** | *NOPKGDEF* {*NONE* | **'names'**}

Controls which packages or packed objects are to be defined in the WSDOC report. *ONLYPKGDEF* specifies packages or packed objects that are to be the only packages or packed objects defined in the report. *NOPKGDEF* specifies packages or packed objects that are not to be defined in the report. *NOPKGDEF NONE* indicates that all packages and packed objects are to be defined. **names** is a list of package or group names, or names of packed objects; all names must be separated by spaces or commas.

Default: *NOPKGDEF NONE*

Usage: The identifiers named in **names** need not exist in the source file, and need not identify valid packages, groups, or packed objects.

Each group named in **names** is expanded into its composite members. Thus, specifying a group to *ONLYPKGDEF* or *NOPKGDEF* has the same effect as specifying each of the group's members.

You can specify a packed object by following the object's name with a dot and the package's name. For example, *NOPKGDEF 'TRANSFORMS.MATH'* specifies that object *TRANSFORMS* in package *MATH* is not to be defined. *TRANSFORMS.MATH* is a **pathname**; you can form a pathname to a packed object at any level.

You can use *ONLYPKGDEF* with *PAGENOS* and *PAGEFORMAT SINGLE FOR PKGDEF* to update only certain pages of a WSDOC report.

The *QUOTES* and *NOQUOTES* Options

{*QUOTES* | *NOQUOTES*}

Specifies whether quoted strings to the right of execute functions (⍎) are to be analysed for cross-reference tables.

Default: *NOQUOTES*

Usage: The values of the workspace's latent expression (system variable ⎕*LX*) and global setting of ⎕*TRAP* are analysed whether or not *QUOTES* is set.

The *QUADFNS* and *NOQUADFNS* Options

{*QUADFNS* | *NOQUADFNS*}

Specifies whether system functions ("quad functions") are to be included in a WSDOC report's cross-reference tables. (System *variables* are always included in cross-reference tables.)

Default: *QUADFNS*

The *BRANCHFNS* Option

BRANCHFNS {**'names'** | *NONE*}

Specifies functions you wish to be displayed as branch functions in the tree graph structure diagram. *NONE* indicates that no function is necessarily to be treated as a branch function (it may be anyway, because of the subtree's size—see *TREEHEIGHT*). **names** is a list of function names separated by spaces or commas.

Default: *BRANCHFNS NONE*

Usage: A **branch function** is a function that is called by another function and whose subordinate calling structure is shown in its own tree. A branch function may be a called function whose subordinate structure exceeds the limit set by the *TREEHEIGHT* option; or it may be a function specified by the *BRANCHFNS* option. A branch function appearing as part of a tree is marked with ⊛ 'log' after the function name. In the branch function's subtree itself, the name of the root is prefixed with ⊛.

For more information about the tree graph structure diagram, see "The *WSGRAPH* Option".

The *TREEROOTS* Option

TREEROOTS {*FINDALL*|**'names'**}

Specifies that **names** are to be the only functions displayed as root functions in the WSDOC report's tree graph structure diagram. *FINDALL* indicates that all main functions—and only main functions—are to be treated as root functions.

Default: *TREEROOTS FINDALL*

Usage: A **root function** is a function at the base or root of a function calling tree. Unless you reset *TREEROOTS*, root functions are the main functions in the workspace—that is, those not called by any other functions.

For more information about the tree graph structure diagram, see "The *WSGRAPH* Option".

The *SURROGATES* Option

SURROGATES {'[*B*][*C*][*I*][*L*][*N*]'|*NONE*}

Specifies which non-typeable characters are to be displayed as surrogates in variable and package definitions. Surrogates are formed by overstriking the single-character mnemonic (for example, *B* for BACKSPACE) with the dieresis character, ¨. These are the characters you can specify to *SURROGATES*:

Mnemonic	Character
B	BACKSPACE
C	CARRIAGE RETURN
I	IDLE
L	LINEFEED
N	NULL

Default: *SURROGATES 'BILN'*. This preserves RETURNS (so that text vectors containing them print properly), but other non-typeable characters are translated to surrogates. For example, if *LF* and *BS* are the linefeed and backspace characters, respectively, then a variable formed by the expression *'LINE',LF,'FEED AND BACK',BS,'SPACE'* would display as *LINEÏFEED AND BACKB̈SPACE* with the default *SURROGATES* setting.

The *XREFSYMBS* Option

XREFSYMBS {**'symbols'**|*NONE*}

Specifies that **symbols** are to be included in the WSDOC report's cross-reference tables. **symbols** is a list of SHARP APL characters, with no blanks or other internal delimiting characters.

Default: *XREFSYMBS* '→⍎:⎕⍞'

Chapter 8

WSDOC REPORT ELEMENTS

A WSDOC report is made up of **report elements** that you can select or suppress. To select an element, you specify it to WSDOC's *LIST* or *LISTONLY* option; to suppress an element, you specify it to the *NOLIST* or *NOLISTONLY* option. (For more information, see "The *LIST*, *NOLIST*, *LISTONLY*, and *NOLISTONLY* Options".)

The following table shows the order in which the report elements appear, and the *LIST* option that specifies them. "*" indicates that WSDOC begins a new page before it begins the starred element.

	LIST **option**	**Report element**
*	*BANNERS*	Leading banner page
*	*COVER*	Cover
*	*TBLCONT*	Table of contents
*		Workspace environment display—controlled by *WSENV* (*default*)
	FNS	Function name display *(default)*
	VARS	Variable name display *(default)*
	GRPS	Group name display *(default)*
	GRP	Group definitions—)*GRP* display for every group *(default)*
*	*FNSUM*	Function summary *(default)*; or
	FNSUMC	Function summary with syntax and comments
*	*VARSUM*	Variable summary *(default)*
*	*GRPSUM*	Group summary *(default)*
*	*PKGSUM*	Package summary *(default)*
*	*FNDEF*	Function definitions *(default)*
	FNXREF	Function cross-reference tables—interleaved with function definitions, if both selected
*	*VARDEF*	Variable definitions *(default)*
*	*PKGDEF*	Package definitions
*	*WSXREF*	Workspace cross-reference table
*	*FNCALLS*	Function call listing
*	*WSGRAPH*	Tree graph structure
*	*BANNERS*	Trailing banner page

You can also use the *LIST* option to control the printing of informative messages that the *WSDOC* program issues.

The *BANNERS* Option

Select *BANNERS* to produce leading and trailing banner pages. The default setting is *NOLIST BANNERS*.

The leading banner page contains the words "WSDOC FOR" followed by the library number and the name of the workspace, in large letters. The trailing banner page contains the same information as the leading banner page, but in reverse contrast. Both occupy a full page (defined by the specified *WIDTH* and *DEPTH* settings).

The *COVER* Option

Select *COVER* to produce a cover (title page) for the WSDOC report. This cover includes the information given at the top of each page (that is, title, workspace name, and workspace time-stamp); and also includes when the WSDOC report was generated. The default setting is *NOLIST COVER*.

Below is an example of a cover from a WSDOC report.

```
************************************
*                                  *
*     APL CHESS PLAYING PROGRAMS   *
*                                  *
*                                  *
*    WORKSPACE DOCUMENTATION FOR   *
*           1234567 CHESS          *
*                                  *
*    VERSION   17.54.28 06/12/84   *
*                                  *
*                                  *
*  TUE 12 JUNE 1984        6:34 PM *
*                                  *
************************************
```

The *TBLCONT* Option

Select *TBLCONT* to produce a table of contents for the WSDOC report. The table of contents gives every report element included in the report, and the number of the page on which it begins. The default setting is *NOLIST TBLCONT*.

The *FNS* Option

Select *FNS* to display the names of the functions in the source file. This display resembles the output of the SHARP APL system command *)FNS*. The default setting is *LIST FNS*.

The *VARS* Option

Select *VARS* to display the names of the variables in the source file. This display resembles the output of the SHARP APL system command *)VARS*. The default setting is *LIST VARS*.

The *GRPS* Option

Select *GRPS* to display the names of the groups in the source file. This display resembles the output of the SHARP APL system command *)GRPS*. The default setting is *LIST GRPS*.

The *GRP* Option

Select *GRP* to display group definitions—that is, the names contained in each group in the source file. This display resembles the output of the SHARP APL system command *)GRP*. The default setting is *LIST GRP*.

Shown below is an example of the display produced when the *FNS*, *VARS*, *GRPS*, and *GRP* options are selected.

```
      )FNS
ADJUSTKF        ALPHABETA       CENTRALITY∆SCORE        CHESS   CRANK   DETERMINE∆SIDES DISPLAY DMATE   END∆HEURISTIC∆SCORE
GAMELDONE       GETMOVE GETSQ   HEURISTIC∆SCORE HRECORD INCHECK INPUT∆AND∆MAKE  INTMOVE KCAPS   KING∆ATTACK∆SCORE       KMOVE
LEGALLCAPTURES  LEGALLMOVES     MACHINESPOINTS  MACHINES∆MOVE   MAKE    MATE    MOBILITY∆SCORE  NCAPS   NEWGAME NMOVE
NOTATIONLFOR    ON      OPPONENTSPOINTS OPPONENTS∆MOVE  OTHER   OUTPUT∆AND∆MAKE PAWN∆PROMOTION∆SCORE    PCAPS   PMOVE   QI
QRBCAPS QRBMOVES        RANK    RBL     RCAT    RECORD  SETC    SETUP   SET∆KING∆FIELD  SQUAREID        SQZ     STALE∆MATE
STATIC  SUPPORTS        TEST∆MOVE       TOWARDS∆CENTRE  UPDATEWEIGHTS

      )VARS
BCATA   BINC    BISHOP  BLACK   BOARD   CMPINC  CPULIM  DMPINC  ENDKFF  ENDPR   FILEM   FILEMR  INITVALS        KFV     KINC
KING    KNIGHT  KEE     LE      MIDHSCORE       MIDPR   NINC    PAWN    PCINC   PIECES  QINC    QUEEN   RINC    ROOK    SMPINC
WHITE   WV

      )GRPS
UTIL

      )GRP UTIL
QI      RCAT    SQZ
```

The *FNSUM* and *FNSUMC* Options

Select *FNSUM* or *FNSUMC* to produce the function summary: a table summarizing the functions in the source file. *FNSUM* and *FNSUMC* are mutually exclusive: if you select one, the other is suppressed. If both are selected, *FNSUMC* is chosen. The default setting is *LIST FNSUM ◇ NOLIST FNSUMC*.

If you select *FNSUM*, the table gives each function's name, type, number of lines, and size in bytes. The type is one of:

ED	explicit dyadic function
EM	explicit monadic function
EN	explicit niladic function
ND	nonexplicit dyadic function
NM	nonexplicit monadic function
NN	nonexplicit niladic function
⍒	locked function

The *FNSUM* display is formatted in as many columns as will fit across the page.

An example of a *FNSUM* display is shown below.

```
APL CHESS PLAYING PROGRAMS                                                                                    PAGE  1
1234567 CHESS * SAVED  17.54.28 06/12/84

FUNCTION NAME              TYPE   LINES   SIZE        FUNCTION NAME              TYPE   LINES   SIZE

ADJUSTKF                   NN       2     148         NMOVE                      EN       3     216
ALPHABETA                  EN      35   1,932         NOTATION∆FOR               ED      15     924
CENTRALITY∆SCORE           EN       2     196         ON                         ED       1      96
CHESS                      NN       4     264         OPPONENTSPOINTS            EN       1     116
CRANK                      EM       2     144         OPPONENTS∆MOVE             EN      10     536
DETERMINE∆SIDES            NN       2     148         OTHER                      EM       2     132
DISPLAY                    NN       3     252         OUTPUT∆AND∆MAKE            NM      16     944
DMATE                      EM       9     488         PAWN∆PROMOTION∆SCORE       EN       2     184
END∆HEURISTIC∆SCORE        EN       1      68         PCAPS                      EN       3     196
GAME∆DONE                  NN       4     248         PMOVE                      EM       6     424
GETMOVE                    EN       8     584         QI                         ED       3     228
GETSQ                      EM      18   1,160         QRBCAPS                    ED       4     228
HEURISTIC∆SCORE            EN      23   1,936         QRBMOVE3                   ED       5     312
HRECORD                    EN       3     348         RANK                       EM       1      96
INCHECK                    EM       3     188         RBL                        EM       2     124
INPUT∆AND∆MAKE             NM       1      84         RCAT                       ED       2     180
INTMOVE                    ED      18   1,080         RECORD                     NM       3     172
KCAPS                      EN       3     176         SETC                       ED       2     148
KING∆ATTACK∆SCORE          EN       1      80         SETUP                      NN      20     812
KMOVE                      EN       3     172         SET∆KING∆FIELD             NM       3     160
LEGALLCAPTURES             EM      11     700         SQUAREID                   ED       3     236
LEGALLMOVES                EM      19   1,148         SQZ                        EM       3     248
MACHINESPOINTS             EN       1     116         STALE∆MATE                 EM       2     152
MACHINES∆MOVE              EN      22   1,048         STATIC                     EN       1      84
MAKE                       NM      12     812         SUPPORTS                   ED       3     192
MATE                       EM       2     152         TEST∆MOVE                  ED       7     592
MOBILITY∆SCORE             EN       3     252         TOWARDS∆CENTRE             EN       1      88
NCAPS                      EN       3     196         UPDATEWEIGHTS              NN       1     108
NEWGAME                    NN       5     296

                      TOTAL (57 FNS) - 22,344
```

If you select *FNSUMC*, the table gives:

- each function's name and syntax (in the form **z←x fn y**);
- number of lines;
- size in bytes; and
- first or last line, if the line is a full-line comment.

If the function is locked, its name is enclosed by parentheses.

An example of a *FNSUMC* display follows.

```
APL CHESS PLAYING PROGRAMS                                                                                    PAGE  2
1234567 CHESS * SAVED  17.54.28 06/12/84

      NAME AND SYNTAX              LINES   SIZE   COMMENTS

        ADJUSTKF                     2     148   ⍝ ADJUST KING FIELD IF LAST MOVE WAS K-?.
      Z←ALPHABETA                   35   1,932   ⍝ ALPHA-BETA CHESS HEURISTIC DRIVER.
      Z←CENTRALITY∆SCORE             2     196   ⍝ COMPUTE SCORE VALUES FOR PIECE CENTRALITY.
        CHESS                        4     264
      Z←CRANK RANK                   2     144   ⍝ COMPUTE BOARD RANK BASED ON PLAYER'S COLOUR.
        DETERMINE∆SIDES              2     148
        DISPLAY                      3     252   ⍝ DISPLAY THE CHESS BOARD.
      Z←DMATE COLOUR                 9     488   ⍝ DETERMINE IF PLAYER IS IN A MATE SITUATION.
      Z←END∆HEURISTIC∆SCORE          1      68
        GAME∆DONE                    4     248   ⍝ DISPLAY FINAL END-OF-GAME PARTINGS.
      Z←GETMOVE                      8     584   ⍝ ACQUIRE PLAYER'S NEXT MOVE.
      Z←GETSQ S                     18   1,160   ⍝ GET THE BOARD POSITIONS OF THE SQUARES IDENTIFIED BY <S>.

          .                          .
          .                          .
          .                          .

Z←COLOUR SQUAREID SQ                 3     236   ⍝ COMPUTE A CHARACTER SQUARE IDENTIFICATION RELATIVE TO A PARTICULAR PLAYER.
      Z←SQZ X                        3     248   ⍝ REMOVES ALL LEADING, TRAILING, AND MULTIPLE INTERNAL BLANKS FROM <X>.
      Z←STALE∆MATE COLOUR            2     152   ⍝ RETURN  1  IF THE SPECIFIED PLAYER IS IN STALE MATE.
      Z←STATIC                       1      84
Z←COLOUR SUPPORTS SQUARES            3     192
  Z←TEST TEST∆MOVE M                 7     592   ⍝ MAKE THE MOVE <M>, PERFORM <TEST>, AND THEN RESCIND THE MOVE.
      Z←TOWARDS∆CENTRE               1      88
        UPDATEWEIGHTS                1     108

                      TOTAL (57 FNS) - 22,344
```

The *VARSUM* Option

Select *VARSUM* to produce the variable summary: a table summarizing the variables in the source file. The default setting is *LIST VARSUM*.

The table gives each variable's name, type, rank, size, and shape. A variable's type is one of:

AR	nested array
BL	Boolean
CH	character
CM	complex
FL	floating-point
IN	integer
PK	package

The variable summary is formatted in as many columns as will fit across the page.

An example of a variable summary is shown below.

```
APL CHESS PLAYING PROGRAMS
1234567 CHESS * SAVED  17.54.28 06/12/84                                                                    PAGE  4
```

VARIABLE NAME	TYPE	RANK	SIZE	SHAPE	VARIABLE NAME	TYPE	RANK	SIZE	SHAPE
BDATA	BL	2	132	128 7	KNIGHT	IN	0	16	10
BINC	IN	2	132	4 7	KER	IN	2	44	2 3
BISHOP	IN	0	16	10	LF	CH	0	16	10
BLACK	IN	0	16	10	MIDHSCORE	IN	1	88	18
BOARD	IN	1	272	64	MIDER	IN	1	40	6
CHECKING	IN	1	24	2	MINC	IN	1	48	8
CPCLINK	IN	1	40	6	PAWN	BL	0	16	10
CHECKINC	IN	0	16	10	PCINC	IN	2	36	2 2
ENDKEP	IN	1	212	49	PIECES	CH	1	32	13
ENDER	IN	1	40	6	QINC	IN	2	244	8 7
FILEN	CH	2	92	24 3	QUEEN	IN	0	16	10
FILEMP	CH	2	36	8 2	RINC	IN	2	132	4 7
INITIALS	PK	-	2,748	-	ROOK	IN	0	16	10
KEY	IN	1	212	49	SMPINC	IN	0	16	10
KINC	IN	1	48	8	WHITE	BL	0	16	10
KING	IN	0	16	10	WV	IN	1	44	7

```
TOTAL (32 VARS) - 4,872
```

The *GRPSUM* Option

Select *GRPSUM* to produce the group summary: a table summarizing the groups in the source file. The default setting is *LIST GRPSUM*.

The group summary gives each group's name, number of elements, and size in bytes.

The group summary is formatted in as many columns as will fit across the page.

The *PKGSUM* Option

Select *PKGSUM* to produce the package summary: a table summarizing the packages in the source file, and the objects they contain. The default setting is *LIST PKGSUM*.

The table gives each package and packed object's name, type, attributes, size in bytes, and (if the object is an array) shape.

The name of a packed object is given as a pathname. For example, pathname *FFT.TRANSFORMS.MATH* identifies object *FFT*, which is packed in package *TRANSFORMS*, which is

itself packed in package *MATH*. Nesting levels are also highlighted visually by an indent of an increasing number of blanks from the left margin.

Each object's type is one of:

-	name with no referent
AR	nested array
BL	Boolean variable
CH	character variable
CM	complex variable
ED	explicit dyadic function
EM	explicit monadic function
EN	explicit niladic function
FL	floating-point variable
IN	integer variable
ND	nonexplicit dyadic function
NM	nonexplicit monadic function
NN	nonexplicit niladic function
PK	package
⍒	locked function

What appears in the Attribute column depends on the object's class. For packages, the number of members is given; for variables, the rank is given; and for functions, the number of lines is given.

An example of a package summary is shown below.

```
APL CHESS PLAYING PROGRAMS
1234567 CHESS * SAVED  17.54.28 06/12/84                                    PAGE  5

IDENTIFIER NAME                          TYPE   ATTRIBUTE   SIZE   SHAPE

INITVALS                                  PK   MEMBS -  6  2,748
    CENTRE∆CONTROL∆FACTOR.INITVALS        IN   RANK  -  1  1,952   484
    HSCORE.INITVALS                       IN   RANK  -  1     88   18
    KING∆FIELD∆FACTOR.INITVALS            IN   RANK  -  1    212   49
    POSITION.INITVALS                     IN   RANK  -  2    276   2 32
    PRIORITY.INITVALS                     IN   RANK  -  1     40   6
    PWEIGHTS.INITVALS                     IN   RANK  -  1     68   13

                                          TOTAL (1 PKG) - 2,748
```

The *FNDEF* Option

Select *FNDEF* to display the definitions of the functions in the source file. The default setting is *LIST FNDEF*. See page 17 for an example of the display produced when the *FNDEF* option is selected.

The *VARDEF* Option

Select *VARDEF* to display the definitions of the variables in the source file. The default setting is *LIST VARDEF*.

Variables are displayed as they would appear had their names been typed in the workspace being documented. In particular, the workspace's settings of ⎕*PP* and ⎕*PS* are used to produce the final display.

Packages are defined as the canonical form ***PACKAGE*** in the variable definitions. Select *PKGDEF* to display the contents of packages.

An example of the output of the *VARDEF* option is shown below:

```
APL CHESS PLAYING PROGRAMS
1234567 CHESS * SAVED  17.54.28 06/12/84                                                    PAGE  32
                                                                                   VARS: BINC - KNIGHT
      BINC                                                                   FILEM

 ¯23  ¯46  ¯69  ¯92 ¯115 ¯138 ¯161                                          R
  ¯23  ¯46  ¯69  ¯92 ¯115 ¯138 ¯161                                          N
 ¯21  ¯42  ¯63  ¯84 ¯105 ¯126 ¯147                                          B
  ¯21  ¯42  ¯63  ¯84 ¯105 ¯126 ¯147                                          Q
                                                                            K
      BISHOP                                                                B
3                                                                           N
                                                                            R
      BLACK                                                                 QR
2                                                                           QN
                                                                            QB
      BOARD                                                                 QP
316 317 318 319 320 321 322 323 294 295 296 297 298 299 300 301             KP
      272 273 274 275 276 277 278 279 250 251 252 253 254 255               KB
      256 257 228 229 230 231 232 233 234 235 206 207 208 209               KN
      210 211 212 213 184 185 186 187 188 189 190 191 162 163               KR
      164 165 166 167 168 169                                               QRP
                                                                            QNP
      CMPINC                                                                QBP
21 23                                                                       QP
                                                                            KP
      CPULIM                                                                KBP
300 3600 10800 21600 32400 43200                                            KNP
                                                                            KRP
      DMPINC
44                                                                     FILEMR
                                                                            QR
      ENDKFF                                                                QN
4 4 5 6 5 4 4 4 8 10 10 10 8 4 5 10 10 10 10 10 5 6 10 10 10000              QB
      10 10 6 5 10 10 10 10 10 5 4 8 10 10 10 8 4 4 4 5 6 5 4 4              Q
                                                                            K
      ENDPR                                                                 KB
0 4 3 1 1 4                                                                 KN
                                                                            KR

                                                                       INITVALS
                                                                      **PACKAGE**

                                                                       KEY
                                                                      63 64 65 66 67 68 69 41_42 43 44 45_46 47 19 20_21 22 23 24_25
                                                                            ¯3 ¯2 ¯1_0 1_2 3_ ¯25_ ¯24_ ¯23_ ¯22_ ¯21_ ¯20_ ¯19 ¯47 ¯46 ¯45
                                                                             ¯44 ¯43 ¯42 ¯41 ¯69 ¯68 ¯67 ¯66 ¯65 ¯64 ¯63

                                                                       KINC
                                                                      1 ¯1 22 ¯22 21 23 ¯21 ¯23

                                                                       KING
                                                                      6

                                                                       KNIGHT
                                                                      2
```

A single variable may, in some instances, be too large for WSDOC to document. Examples include a very large Boolean array, or a complex nested array being formatted with a value of ⎕*PS* which introduces a lot of space in the result. When WSDOC can't format a variable, the text *** *OBJECT TOO LARGE TO DOCUMENT* *** will appear in the output in place of the variable's definition.

The *PKGDEF* Option

Select *PKGDEF* to display the definitions of the packages in the source file, and of all objects contained in these packages, to all levels. The default setting is *NOLIST PKGDEF*.

The name of a packed object is given as a pathname. For example, pathname *FFT.TRANSFORMS.MATH* identifies object *FFT*, which is packed in package *TRANSFORMS*, which is itself packed in package *MATH*.

Packed functions have associated cross-reference tables printed if *LIST FNXREF* is set, and have entries in the workspace cross-reference table if *LIST WSXREF* is set.

Packed names without referents are indicated by "*VALUE ERROR*" in the package definition display.

An example of the output of *PKGDEF* is shown below:

```
APL CHESS PLAYING PROGRAMS
1234567 CHESS * SAVED  17.54.28 06/12/84                                                
```

A **suspicious identifier** is one whose use is inconsistent with the identifier's class, and is marked in the report with *?*. An identifier is marked suspicious if:

- it is localized but not referenced (except for some system variables);
- it is localized but not assigned (except for some system variables);
- it is localized but defined as a line label;
- it is defined as a line label more than once;
- it is localized more than once;
- it is defined as a line label or global function but is assigned or index-assigned a value;
- it is defined as a line label but is never referenced; or
- it is the explicit result of the function but is never assigned a value.

The **type of appearance** is one of:

BLANK	simple reference or branch
←	simple assignment
[	indexed reference
[←	indexed assignment
:	line-label definition

Refer to page 17 for examples of function cross-reference tables.

It is possible for the width you have specified to be too narrow to accommodate a particular *FNXREF* report, particularly if *PAGEFORMAT TWOCOL FOR FNDEF* was selected. If this occurs, the text **** WIDTH TOO NARROW TO ACCOMMODATE FNXREF **** will appear in place of the particular function cross-reference table.

The *WSXREF* Option

Select *WSXREF* to produce a cross-reference table for the entire workspace. The default setting is *NOLIST WSXREF*.

The workspace cross-reference table has an entry for:

- every system and global variable,
- every global function, and
- every other identifier and every specified symbol that appears in the workspace's functions or packages.

The *XREFSYMBS* option is used to specify APL symbols that are to be cross-referenced. You can use the *ONLYFNDEF* or *NOFNDEF*, and *ONLYPKGDEF* or *NOPKGDEF* options to limit the functions whose internals are to be cross-referenced.

The contents of the workspace's latent expression (system variable *⎕LX*) and global *⎕TRAP* setting are always cross-referenced.

For each identifier or symbol, the workspace cross-reference table gives: the identifier's name; the function or package in which it appears (if the identifier isn't global); its type; whether the identifier is "suspicious"; and the line numbers of the lines on which the identifier appears, with the type of appearance.

In the function column, names of global packages are given as truncated pathnames; for example, *.TRANSFORMS.MATH* means package *TRANSFORMS*, which is contained in package *MATH*. Names of packed functions are given as full pathnames; for example, *FFT.TRANSFORMS.MATH* means function *FFT*, which is packed in package *TRANSFORMS*, which is, in turn, packed in package *MATH*.

For a description of the codes and markings used in the workspace cross-reference table, see "The *FNXREF* Option".

An object is marked suspicious in the *WSXREF* report for the same reasons as in the *FNXREF* report. As well, a global variable is marked suspicious if it is not referenced anywhere in the workspace.

An example of a workspace cross-reference table is shown below.

```
APL CHESS PLAYING PROGRAMS                                                          PAGE  39
1234567 CHESS * SAVED  17.54.28 06/12/84
                                                                  XREF: GAME∆DONE - INCH

IDENTIFIER          FUNCTION            TYPE    REFERENCES

GAME∆DONE                               FN
                    GETMOVE             FN       8
                    OPPONENTS∆MOVE      FN       4
                    OUTPUT∆AND∆MAKE     FN       5    8   15

GAME∆MODE           CHESS               LV  ?    0
                    MACHINES∆MOVE        *       8   13←  17←
                    SETUP                *      18←

GETMOVE                                 FN
                    OPPONENTS∆MOVE      FN       5

GETSQ                                   FN
                    GETSQ               RC      11-
                    INTMOVE             FN       4    6

H                   HEURISTIC∆SCORE     LV       0    1←   6[←  7[←  8[←  9[← 10[← 11[← 12[← 13[← 14[←
                                                16[← 17[← 19[← 20[← 21[← 22[← 23   23

HEURISTIC∆SCORE                         FN
                    MACHINES∆MOVE       FN       9   14

HISTORY             CHESS               LV  ?    0
                    INPUT∆AND∆MAKE       *       1←   1
                    MACHINES∆MOVE        *      10
                    OUTPUT∆AND∆MAKE      *       9←   9
                    SETUP                *      12←

HRECORD                                 FN
                    GAME∆DONE           FN       4

HSCORE              .INITVALS           GV  ?
                    CHESS               LV  ?    0
                    HEURISTIC∆SCORE      *       1   23
                    MACHINES∆MOVE        *      13←

I                   HRECORD             LV       0    1←   2    2
                    INTMOVE             LV       0    3←   4    5    6
                    MOBILITY∆SCORE      LV       0    2←   3    3    3←   3
                    NOTATION∆FOR        LV       0    3←   4    6
                    SET∆KING∆FIELD      LV       0    2←   3

ILL                 INTMOVE             LL       4    6    8   11   14:

IN                  GETMOVE             LV       0    3←   3    4    5    5    7
                    INTMOVE             RA       0    2    2    3    3    4    5[   6   14   15   16

INC                 QRBCAPS             RA       0    2
                    QRBMOVE3            RA       0    2

INCH                INTMOVE             LL      13:  17
```

As with *FNXREF*, it is possible for the width you have specified to be insufficient to accommodate the *WSXREF* report. If this occurs, the message **** WIDTH TOO NARROW TO ACCOMMODATE WSXREF **** will appear in the WSDOC output, and the workspace cross-reference report will be suppressed.

The *FNCALLS* Option

Set *FNCALLS* to produce a function call listing. This is a table of every function in the source file with the functions it calls, in alphabetical order. In effect, it acts as an overview of the tree graph structure diagram. The default setting is *NOLIST FNCALLS*.

If a particular function calls no subfunctions, then "*(NONE)*" will appear opposite its name in the *FNCALLS* listing. Functions which call themselves are highlighted as being recursive by "*(RECUR)*" appearing after the function's name in the call list.

You can use the *ONLYFNDEF* or *NOFNDEF* option to limit the functions that appear in the function call listing.

An example of a function call listing is shown below. Notice that *GETSQ* is a recursive function.

```
APL CHESS PLAYING PROGRAMS
1234567 CHESS * SAVED  17.54.28 06/12/84                                                                                    PAGE  58

FUNCTION NAME            FUNCTIONS CALLED

ADJUSTKF                 SET∆KING∆FIELD
ALPHABETA                INCHECK, LEGAL∆CAPTURES, MAKE
CENTRALITY∆SCORE         (NONE)
CHESS                    DISPLAY, INPUT∆AND∆MAKE, MACHINES∆MOVE, NEWGAME, OPPONENTS∆MOVE, OUTPUT∆AND∆MAKE
CRANK                    (NONE)
DETERMINE∆SIDES          QI
DISPLAY                  (NONE)
DMATE                    LEGAL∆MOVES, SETC, TEST∆MOVE
END∆HEURISTIC∆SCORE      (NONE)
GAME∆DONE                DISPLAY, HRECORD
GETMOVE                  DISPLAY, GAME∆DONE, INTMOVE, QI, RBL, RECORD, SQZ
GETSQ                    CRANK, GETSQ (RECUR)
HEURISTIC∆SCORE          ON, OTHER, STATIC, SUPPORTS, TOWARDS∆CENTRE
HRECORD                  (NONE)
INCHECK                  OTHER, SUPPORTS
INPUT∆AND∆MAKE           ADJUSTKF, MAKE
INTMOVE                  GETSQ, INCHECK, OTHER, TEST∆MOVE
KCAPS                    (NONE)
KING∆ATTACK∆SCORE        (NONE)
KMOVE                    (NONE)
LEGAL∆CAPTURES           KCAPS, NCAPS, PCAPS, QRBCAPS
LEGAL∆MOVES              KMOVE, NMOVE, OTHER, PMOVE, QRBMOVE3, SUPPORTS
MACHINESPOINTS           (NONE)
MACHINES∆MOVE            ALPHABETA, CENTRALITY∆SCORE, END∆HEURISTIC∆SCORE, HEURISTIC∆SCORE, KING∆ATTACK∆SCORE, LEGAL∆MOVES,
                         MACHINESPOINTS, MOBILITY∆SCORE, OPPONENTSPOINTS, PAWN∆PROMOTION∆SCORE, SET∆KING∆FIELD, STATIC, UPDATEWEIGHTS
MAKE                     OTHER, SETC
MATE                     DMATE, INCHECK
MOBILITY∆SCORE           LEGAL∆MOVES, MAKE
NCAPS                    (NONE)
NEWGAME                  DETERMINE∆SIDES, QI, SETUP
NMOVE                    (NONE)
NOTATION∆FOR             INTMOVE, LEGAL∆MOVES, OTHER, SQUAREID

      .                     .
      .                     .
      .                     .

SETC                     (NONE)
SETUP                    (NONE)
SET∆KING∆FIELD           (NONE)
SQUAREID                 CRANK, RBL
SQZ                      (NONE)
STALE∆MATE               DMATE, INCHECK
STATIC                   (NONE)
SUPPORTS                 LEGAL∆MOVESTEST∆MOVE              SETC
TOWARDS∆CENTRE           (NONE)
UPDATEWEIGHTS            MACHINESPOINTS, OPPONENTSPOINTS
```

The *WSGRAPH* Option

Select *WSGRAPH* to produce a tree graph **structure diagram** of functions in the WSDOC source file. The default setting is *NOLIST WSGRAPH*.

The tree graph structure diagram shows, in the form of a tree, the function-calling hierarchy of the workspace.

The function at the base of each tree—the **root function**—is either a function not called by any other, or a function specified to the *TREEROOTS* option. You can use the *ONLYFNDEF* or *NOFNDEF* option to limit the set of functions that WSDOC searches for root functions.

A **branch function** is a called function whose subtree is broken out to form its own tree. A subtree is a called function's own subordinate hierarchy of function calls. Branch functions are marked with ⊛. A called function becomes a branch function if:

- it is specified to the *BRANCHFNS* option; or
- its subtree's height exceeds the value of the *TREEHEIGHT* option.

A **recursive function** is a function that calls itself, directly or indirectly. Recursion is indicated by enclosing the recursive function's name in parentheses.

If a tree is too wide to fit on a single page, WSDOC prints it in segments that you can fit together manually if desired.

An example of a tree graph structure diagram follows.

```
APL CHESS PLAYING PROGRAMS
1234567 CHESS * SAVED  17.54.28 06/12/84                                                    PAGE  60

                                                                                  TREE: ALPHABETA - INTMOVE

                                  ⌈- OTHER
             ⌈- INCHECK --------|                                ⌈- KMOVE
             |                  |                                |- NMOVE
             |                  L- SUPPORTS --- LEGAL∆MOVES -|- OTHER
             |                                                  |- PMOVE
             |                                                  |- QRBMOVE3
⊛ ALPHABETA -|                                                  L- (SUPPORTS)
             |                  ⌈- KCAPS
             |- LEGAL∆CAPTURES -|- NCAPS
             |                  |- PCAPS
             |                  L- QRBCAPS
             |
             L- MAKE -----------⌈- OTHER
                                L- SETC


         ⌈- DISPLAY
         |- INPUT∆AND∆MAKE --⌈- ADJUSTKF ---------- SET∆KING∆FIELD
         |                   |
         |                   L- MAKE ------------⌈- OTHER
         |                                       L- SETC
- CHESS -|- MACHINES∆MOVE ⊛
         |                   ⌈- DETERMINE∆SIDES --- QI
         |- NEWGAME ---------|- QI
         |                   L- SETUP
         |- OPPONENTS∆MOVE ⊛
         L- OUTPUT∆AND∆MAKE ⊛


           ⌈- GETSQ -----⌈- CRANK
           |             L- (GETSQ)
           |
           |             ⌈- OTHER
           |- INCHECK ---|                          ⌈- KMOVE
           |             |                          |- NMOVE
⊛ INTMOVE -|             L- SUPPORTS --- LEGAL∆MOVES -|- OTHER
           |                                        |- PMOVE
           |                                        |- QRBMOVE3
           |                                        L- (SUPPORTS)
           |- OTHER
           L- TEST∆MOVE --- SETC
```

The *MSGS* Option

Set *MSGS* to have WSDOC display messages about the WSDOC job before, during, and after processing. The default setting is *LIST MSGS*.

If you set *NOPRINT* but allow *TTASK* and *LIST MSGS*, WSDOC displays which report element it is processing as it begins that element.

The *SYSCMDS*, *SUMS*, *DEFS*, and *XREFS* Options

You can use these elements to select several report elements at once. For example, you can set *NOLIST DEFS* to suppress all object definitions.

Multiple element	Elements selected
SYSCMDS	*FNS*, *VARS*, *GRPS*, *GRP*
SUMS	*FNSUM*, *VARSUM*, *GRPSUM*, *PKGSUM*
DEFS	*FNDEF*, *VARDEF*, *PKGDEF*
XREFS	*FNXREF*, *WSXREF*

The *ALL* and *NONE* Options

LIST ALL, *LISTONLY ALL*, or *NOLISTONLY NONE*
Selects every report element, including *FNSUMC* and excluding *FNSUM*.

NOLIST ALL, *NOLISTONLY ALL*, or *LISTONLY NONE*
Suppresses every report element. The workspace environment display may still be produced, depending on the setting of the *WSENV* option.

LIST NONE and *NOLIST NONE*
Have no effect.

Chapter 9

WSDOC PROFILES

You can use the SHARP APL Workspace Profile Facility—PROFILE—to set up a profile for workspace 7 *STATE* or 7 *WSDOC*. A workspace's profile is a program that is automatically executed when you load the workspace. Both the *STATE* and the *WSDOC* workspaces contain simple default profiles.

Set a *STATE* profile when you find yourself setting the same options for all or nearly all your WSDOC jobs. Because the *WSTOFILE* program's last action is to load 7 *STATE*, the *STATE* profile is executed whenever you execute *WSTOFILE*.

Set a *WSDOC* profile when you want a program executed before or after the *WSDOC* program. An example might be a package interfaced to WSDOC, requiring automatic notification of the completion of the processing of a report.

You use the *SETPROFILE* program from workspace 7 *PROFILE* to set and reset profiles. A workspace profile may comprise any number of functions and variables, all of which you specify in a namelist argument to *SETPROFILE*. The profile need not include any functions, nor, in fact, any objects. If, however, it includes a function with the name *PROFILE*, then that function will be executed automatically whenever the appropriate workspace is loaded.

The global variable *PROFID* in each of the *STATE* and *WSDOC* workspaces identifies to the profile facility the application workspace which has been activated. In 7 *STATE*, *PROFID* is *WSDOC STATE*; in 7 *WSDOC*, it is simply *WSDOC*.

The *STATE* workspace's **default profile** executes the *WSDOCSTATUS* program, which displays the status of any N- or B-task WSDOC jobs you've started or that have ended since you last executed *WSDOCSTATUS*. The *WSDOC* workspace's default profile executes the *WSDOC* program.

To set or reset a profile:

1. Make sure your active workspace is the one you want to set a profile for.
2. Copy *SETPROFILE* into the active workspace.
3. Redefine or define the objects that are to form the profile.
4. Execute *SETPROFILE*.

Example. The user sets his *STATE* profile. His profile includes not only a redefined *PROFILE* function but a character string, *T*, for use in titles, and a character scalar, *CR*, for the RETURN character.

```
      )WSID
IS 7 STATE
      ⎕LX
GETPROFILE
      PROFID
WSDOC STATE
      )COPY 7 PROFILE SETPROFILE
SAVED  16.54.09 04/23/84
      ∇PROFILE[⎕]
    ∇ PROFILE
[1]   WSDOCSTATUS
    ∇
[2]   NTASK
[3]   CREATEOUT
[4]   LIST FNSUMC, XREFS
[5]   ∇
      T←'BILL SMITH',CR←⎕AV[⎕IO+156]
      SETPROFILE 'PROFILE CR T'
<PROFILE> RESET
```

Chapter 10

WSDOC UTILITY PROGRAMS

The *STATE* workspace includes three utility programs:

GENWSTOFILE
MRGWSTOFILES
WSDOCSTATUS

The *GENWSTOFILE* Program

pgmfile [**srcfile**] *GENWSTOFILE* **comp**
Generates a WSDOC source file from all or part of a free-format program file.

pgmfile is the tie number of the program file. **srcfile** is the optional tie number of an existing WSDOC source file into which **pgmfile** is to be merged; if you omit **srcfile**, a new source file is created.

comp specifies the components of the program file to be inspected. If **comp** is a non-empty vector, it indicates that only the specified components are to be inspected. If **comp** is a scalar, it indicates that components from **comp** through the last are to be inspected. If **comp** is an empty vector, it indicates that every component is to be inspected.

Usage: *GENWSTOFILE* is particularly useful when documenting a database of functions and variables too large to simultaneously fit into a workspace.

The object directories in the generated source file are not sorted. Set the WSDOC state *ALPHASORT* option to sort them.

The input program file may contain any type of data. Components that conform to the requirements of 1 ⎕*FD* ω or 2 ⎕*FD* ω are defined directly into the source file. Packages are defined one level deep, and the objects thus formed are added to the source file as functions or variables.

Examples. Generate a source file from a program file:

```
        '588 SORCERER' ⎕STIE 1
        1 GENWSTOFILE ''
FUNCTION FILE 1 - 588 SORCERER
SRCFILE NAME OR SPACE,CR: SORCERER
SRCFILE 20 - 1234567 SORCERER
215 FUNCTIONS
66 VARIABLES
0 GROUPS
```

Merge a program file (tied to 2) with a source file (tied to 20); use components 10 on:

```
        2 20 GENWSTOFILE 10
FUNCTION FILE 2 - 588 NECROMANCER
SRCFILE 20 - 1234567 SORCERER
332 FUNCTIONS
```

```
82 VARIABLES
0 GROUPS
```

Merge a program file with a source file, but inspect only those components of the program file specified:

```
      3 20 GENWSTOFILE 2 5 7 9 12 13 14 23 28
```

The *MRGWSTOFILES* Program

MRGWSTOFILES **src**

Merges the WSDOC source files whose tie numbers are given in **src**. **src** should be ordered from the most dominant to the least dominant source file. All source files are merged into the first (most dominant) file. Where name conflicts occur, the definition of the object appearing in the more dominant file is retained.

Usage: The object directories in the merged source file are not sorted. Set the WSDOC state *ALPHASORT* option to sort them.

Example:

```
      MRGWSTOFILES 21 20 22
SRCFILE 21 - 1234567 UTILITY - MOST DOMINANT
SRCFILE 20 - 1234567 SETUP
SRCFILE 22 - 1234567 TEST - LEAST DOMINANT
61 FUNCTIONS
73 VARIABLES
4 GROUPS
```

The *WSDOCSTATUS* Program

WSDOCSTATUS

Displays the status of any N- or B-task WSDOC jobs that you've started or requested, or that have ended since the last time you executed *WSDOCSTATUS*. Only status reports on WSDOC jobs run in the last seven days are displayed; others are discarded automatically.

Usage: The default profile of the *STATE* workspace executes *WSDOCSTATUS*.

Example:

```
      WSDOCSTATUS

NTASK TASKID          - 3096
WSDOC FOR             - 1234567 MEGAPROJECT
WSDOC COMPLETED AT    - 23.58.50  14 AUG 1984
HSPRINT REQUEST       - 150256 - FILED

BTASK TASKID          - 6182 (REQUEST NO. 91024)
WSDOC FOR             - 1234567 SORCERER - IN PROGRESS
```

WSDOC SOURCE FILE ORGANIZATION

The source file created by the *WSTOFILE* program contains all the information required by the *WSDOC* program, and includes an image of each function, variable, and group as well as complete environment details and the state indicator stack. The format of the source file is as follows:

Component	Description
1	A package containing the state control variables *CO* and *CHARS*. *CO* contains all numeric information that characterizes the state; *CHARS* contains character information such as the title, cross-reference symbols, and special identifier names. If this package contains no objects, the default state is used.
2	Complete information required to submit an *HSPRINT* request for the WSDOC output file; prompted for if the *HIGHSPEED* option was set and this component is empty. The information includes the print specifications and the disposition of the output.
3-7	Not used.
8	Event report, *⎕ER*.
9	Event traps, *⎕TRAP*.
10	Latent expression, *⎕LX*.
11	Control information for the workspace. This is a three-element vector giving the space used by the workspace, its owner, and the timestamp that specifies when it was last saved.
12	Workspace identifier, *)WSID*.
13	State indicator stack, *)SI*.
14	Workspace environment vector. This includes origin, digits, width, comparison tolerance, random link, workspace available, symbols reserved, symbols used, array positioning and spacing, and environment condition.
15	Names of the functions in the workspace, *)FNS*.
16	Bytes used by each function.
17	Index to the components corresponding to each function.
18	Names of the variables in the workspace, *)VARS*.
19	Bytes used by each variable.
20	Index to the components corresponding to each variable.
21	Names of the groups in the workspace, *)GRPS*.
22	Bytes used by each group.
23	Index to the components corresponding to each group.
24-	Start of the components associated with the contents of each group, the listing of each function, and the value of each variable in the workspace. These components are indexed by components 17, 20, and 23.

During the execution of *WSDOC*, the source file is used as a temporary storage file for the banner page if *BANNERS* was selected, and for information gathered during the cross-reference analysis phase if any of *FNXREF*, *WSXREF*, *FNCALLS*, or *WSGRAPH* was selected. Information concerning references within each function is appended to the file when workspace buffers become too full. These temporary components are removed from the file at the completion of the WSDOC, thus restoring the file to its previous condition.

Appendix B

WSDOC STATE CONTROL VECTORS

WSDOC uses two state control vectors to specify the state. These two vectors, *CQ* and *CHARS*, are stored as a package in component 1 of the source file, and are manipulated by the functions in workspace 7 *STATE*. Each vector is described below. The default value of each element is given in parentheses after its description.

Numeric State Control Vector, *CQ*

Element	Description
1	Page width (132)
2	Page depth (66)
3	Top margin (2)
4	Bottom margin (4)
5	Left margin (0)
6	Right margin (0)
7	Task type: 0 for *TTASK*; 1 for *NTASK*; 2 for *BTASK* (0)
8	Print output on the terminal: 0 for *NOPRINT*; 1 for *PRINT* (1)
9	Submit *HSPRINT* request: 0 for *NOHIGHSPEED*; 1 for *HIGHSPEED* (0)
10	Display timestamps for identifiers: 0 for *NOSTAMP*; 1 for *STAMP* (0)
11	Display statements with diamonds on separate lines: 0 for *NODIAMOND*; 1 for *DIAMOND* (0)
12	Disposition of source file: 0 for *SAVESRC*; 1 for *ERASESRC* (1)
13	Output file number: ¯1 for *CREATEOUT*; 0 for no output file; natural number for output file tie number (0)
14	Keyboard message state: 0 for *KEYB NOMSG*; 1 for *KEYB NORMAL* (0)
15	Page pause for realignment: 0 for no pause; natural number for number of pages to print before pausing for alignment of paper (3)
16	Negative of the number of page numbers to use (the actual page numbers are stored in elements 56+⍳-*CQ*[16]): 0 for no numbering; ¯1 to increment by 1 from *CQ*[57]; <¯1 to use only the numbers *CQ*[56+⍳-*CQ*[16]] (¯1)
17	Print summaries: 0 for *ALPH*; 1 for *ASCENDING*; ¯1 for *DESCENDING* (0)
18	Page format for *FNDEF*: 0 for *RUNNING*, 2 for *SINGLE*, 4 for *MULTIPLE* (4); and 0 for *ONECOL*, 1 for *TWOCOL* (0); state value is the sum of the settings for each category (4)
19	Page format for *VARDEF*: 0 for *RUNNING*, 2 for *SINGLE*, 4 for *MULTIPLE* (4); and 0 for *ONECOL*, 1 for *TWOCOL* (0); state value is the sum of the settings for each category (4)
20	Page format for *PKGDEF*: 0 for *RUNNING*, 2 for *SINGLE*, 4 for *MULTIPLE* (4); and 0 for *ONECOL*, 1 for *TWOCOL* (0); state value is the sum of the settings for each category (4)
21	Workspace environment: 0 for *NO*; 1 for *FULL*; 2 for *NONDEFAULT* (2)
22	Page format for *FNXREF*: 0 for *RUNNING*, 2 for *SINGLE*, 4 for *MULTIPLE* (4); and 0 for *ONECOL*, 1 for *TWOCOL* (1); state value is the sum of the settings for each category (5)
23	Page format for *WSXREF*: 0 for *RUNNING*, 2 for *SINGLE*, 4 for *MULTIPLE* (4); and 0 for *ONECOL*, 1 for *TWOCOL* (1); state value is the sum of the settings for each category (5)

24 Selective function control: 0 for *NOFNDEF*; 1 for *ONLYFNDEF* (0)
25 Selective variable control: 0 for *NOVARDEF*; 1 for *ONLYVARDEF* (0)
26 Selective package control: 0 for *NOPKGDEF*; 1 for *ONLYPKGDEF* (0)
27 Tree height (10)
28 Display system functions in cross-references: 0 for *NOQUADFNS*; 1 for *QUADFNS* (1)
29 Alphabetically sort categories within the source file. State value is the sum of the appropriate powers of two: 0 for *NONE*; 1 for *FNS*; 2 for *VARS*; 4 for *GRPS*; 8 for *GRP*; 16 for *PKGS* (24)
30 Title centering: 0 for no centering; 1 for centre (1)
31 Scan executed quoted strings during cross-referencing: 0 for *NOQUOTES*; 1 for *QUOTES* (0)
32 Replace non-typeable characters with surrogates in the display of variables; state value is the sum of the appropriate powers of two: 0 for none; 1 for BACKSPACE; 2 for RETURN; 4 for IDLE; 8 for LINEFEED; 16 for NULL (29)
33 Not used (¯1)
34 Not used (¯1)
35 *BANNERS*: 0 for *NOLIST*; 1 for *LIST* (0)
36 *COVER*: 0 for *NOLIST*; 1 for *LIST* (0)
37 *TBLCONT*: 0 for *NOLIST*; 1 for *LIST* (0)
38 *FNS*: 0 for *NOLIST*; 1 for *LIST* (1)
39 *VARS*: 0 for *NOLIST*; 1 for *LIST* (1)
40 *GRPS*: 0 for *NOLIST*; 1 for *LIST* (1)
41 *GRP*: 0 for *NOLIST*; 1 for *LIST* (1)
42 *FNSUM*: 0 for *NOLIST*; 1 for *LIST* (1)
43 *FNSUMC*: 0 for *NOLIST*; 1 for *LIST* (0)
44 *VARSUM*: 0 for *NOLIST*; 1 for *LIST* (1)
45 *GRPSUM*: 0 for *NOLIST*; 1 for *LIST* (1)
46 *PKGSUM*: 0 for *NOLIST*; 1 for *LIST* (1)
47 *FNDEF*: 0 for *NOLIST*; 1 for *LIST* (1)
48 *VARDEF*: 0 for *NOLIST*; 1 for *LIST* (1)
49 *PKGDEF*: 0 for *NOLIST*; 1 for *LIST* (0)
50 *FNXREF*: 0 for *NOLIST*; 1 for *LIST* (0)
51 *WSXREF*: 0 for *NOLIST*; 1 for *LIST* (0)
52 *FNCALLS*: 0 for *NOLIST*; 1 for *LIST* (0)
53 *WSGRAPH*: 0 for *NOLIST*; 1 for *LIST* (0)
54 *MSGS*: 0 for *NOLIST*; 1 for *LIST* (1)
55 Not used (¯1)
56 Not used (¯1)
57- Page numbers (1)

Character State Control Vector, *CHARS*

The variable *CHARS* holds character state information, excluding the *HSPRINT* delivery instructions, which are kept in component 2 of the source file.

CHARS is a partitioned character vector whose default value is the sequence 1⍬(⍬1⍬¯8↑⎕*AV*),'→⍎:⎕⎕'. This includes a series of eight partition delimiters from ⎕*AV*, and the default cross-reference symbol list. Partitions in *CHARS* are stored in the following order:

1 Title
2 Selective function control
3 Selective variable control
4 Selective package control
5 Tree root names
6 Branch functions
7 Cross-reference symbol list

WSDOC STATE OPTION SUMMARY

Syntax. Square brackets—[ ]—indicate that an element is optional. Braces and staves indicate that you must choose only one element from a list; for example, *{NO|NONDEFAULT|FULL} WSENV* indicates that you must enter *NO WSENV* or *NONDEFAULT WSENV* or *FULL WSENV*.

ALPHASORT {[*FNS*][,*VARS*][,*GRPS*][,*GRP*][,*PKGS*]|*ALL*|*NONE*}
Format option; specifies which classes of identifiers in the WSDOC source file are to be sorted alphabetically before the file is analysed. Default: *ALPHASORT GRP, PKGS*

BRANCHFNS {**'names'**|*NONE*}
Content option; specifies functions that are necessarily to be displayed as branch functions in the tree graph structure diagram. Default: *BRANCHFNS NONE*

BTASK
Control option; indicates that the WSDOC job is to run in a B-task. Overrides *TTASK* (the default) or *NTASK*.

CREATEOUT
File option; indicates that WSDOC is to create the output file. Overrides *OUTFILE* (the default).

DEFAULT
State management function; restores the default state.

DEPTH **n**
Format option; specifies that the WSDOC report's pages are to be **n** lines deep. Default: *DEPTH* 66

DIAMOND
Format option; specifies that function lines in function definitions are to be reformatted; that is, that function lines that contain two or more statements (separated by ◇ 'diamond') are to be shown with each statement starting on a new line; and that the start of each statement (after any label) is to be indented past the longest label, so that each statement starts in the same column. Overrides *NODIAMOND* (the default).

ERASESRC
File option; specifies that the source file is to be erased when the WSDOC report is done. This is the default setting. Overrides *SAVESRC*.

pgmfile [**srcfile**] *GENWSTOFILE* **comp**
Utility program; generates a WSDOC source file from all or part of a free-format function file.

GETSTATE {**filename**|**tieno**}
State management function; replaces the current state with the state stored in the specified WSDOC source file.

HIGHSPEED
File option; indicates that WSDOC is to prepare and submit an *HSPRINT* request for the output file when the report is done. Overrides *NOHIGHSPEED* (the default).

HSPRINT
File option; changes *PRINT* to *NOPRINT*, *NOHIGHSPEED* to *HIGHSPEED*, and *OUTFILE* 0 to *CREATEOUT*.

KEYB {NOMSG|NORMAL}
Control option; specifies whether your terminal can receive messages while the WSDOC report is being composed. Default: *KEYB NOMSG*.

LIST **elements**
Content option; specifies that **elements** are to be included in the WSDOC report. Default: *LIST FNS*, *VARS*, *GRPS*, *GRP*, *FNSUM*, *VARSUM*, *GRPSUM*, *PKGSUM*, *FNDEF*, *VARDEF*, *MSGS*. Overrides *NOLIST*.

LISTONLY **elements**
Content option; specifies that **elements** are to be the only elements included in the WSDOC report. Overrides *LIST* and *NOLIST*.

MRGWSTOFILES **src**
Utility program; merges the WSDOC source files whose tie numbers are given in **src**.

NODIAMOND
Format option; indicates that function lines in function definitions are not to be reformatted. This is the default setting. Overrides *DIAMOND*.

*NOFNDEF {NONE|['∇'][,]['**names**']}*
Content option; specifies functions that are not to be defined in the WSDOC report. Default: *NOFNDEF NONE*. Overrides *ONLYFNDEF*.

NOHIGHSPEED
File option; indicates that WSDOC is not to prepare or submit an *HSPRINT* request for the output file. This is the default setting. Overrides *HIGHSPEED*.

NOLIST **elements**
Content option; specifies that **elements** are to be excluded from the WSDOC report. Default: *NOLIST BANNERS*, *COVER*, *TBLCONT*, *FNSUMC*, *PKGDEF*, *FNXREF*, *WSXREF*, *FNCALLS*, *WSGRAPH*. Overrides *LIST*.

NOLISTONLY **elements**
Content option; specifies that **elements** are to be the only elements excluded from the WSDOC report. Overrides *LIST* and *NOLIST*.

*NOPKGDEF {NONE|'**names**'}*
Content option; specifies packages or packed objects that are not to be defined in the WSDOC report. Default: *NOPKGDEF NONE*. Overrides *ONLYPKGDEF*.

NOPRINT
Control option; indicates that the WSDOC report is not to be printed at your terminal while it is being composed. Overrides *PRINT* (the default).

NOQUADFNS
Content option; indicates that system functions are not to be included in a WSDOC report's cross-reference tables. Overrides *QUADFNS* (the default).

NOQUOTES
Content option; indicates that quoted strings to the right of execute functions are not to be analysed for cross-reference tables. This is the default setting. Overrides *QUOTES*.

NOSTAMP
Format option; indicates that a timestamp is not to be displayed with each object definition. This is the default setting. Overrides *STAMP*.

NOVARDEF {*NONE*|**'names'**}
Content option; specifies variables that are not to be defined in the WSDOC report. Default: *NOVARDEF NONE*. Overrides *ONLYVARDEF*.

NTASK
Control option; indicates that the WSDOC job is to be run in an N-task. Overrides *TTASK* (the default) or *BTASK*.

ONLYFNDEF **'names'**
Content option; specifies functions that are to be the only functions defined in the WSDOC report. Overrides *NOFNDEF* (the default).

ONLYPKGDEF **'names'**
Content option; specifies packages or packed objects that are to be the only such objects defined in the WSDOC report. Overrides *NOPKGDEF* (the default).

ONLYVARDEF **'names'**
Content option; specifies variables that are to be the only variables defined in the WSDOC report. Overrides *NOVARDEF* (the default).

OUTFILE **n**
File option; specifies that the WSDOC report is to be appended to the file tied to **n**. Default: *OUTFILE* 0 (i.e., report not appended). Overrides *CREATEOUT*.

PAGEFORMAT [*SINGLE*|*MULTIPLE*|*RUNNING*] [,*ONECOL*|*TWOCOL*]
[*FOR* [*FNDEF*|,*VARDEF*|,*PKGDEF*|,*FNXREF*|,*WSXREF*|,*DEFS*|,*XREFS*]]
Format option; controls how object definitions and cross-reference tables are to be grouped and formatted on the WSDOC report's pages. Default: *PAGEFORMAT MULTIPLE*, *ONECOL FOR DEFS* ◇ *PAGEFORMAT MULTIPLE*, *TWOCOL FOR XREFS*.

PAGENOS **n**
Format option; specifies either the initial page number, or all page numbers. Default: *PAGENOS* 1.

PAGEPAUSE **n**
Control option; specifies that printing of the WSDOC report is to pause every **n** pages. Effective only when *PRINT* is selected. Default: *PAGEPAUSE* 3.

PRINT
Control option; indicates that the WSDOC report is to be printed at your terminal while it is being composed. This is the default setting. Overrides *NOPRINT*.

PRINTSUMS {ALPH|ASCENDING|DESCENDING}
Format option; specifies the order in which the identifiers in the function, variable, group, and package summaries are to be selected for printing.

PUTSTATE {**tienos**|**fileid**}
State management function; puts the current WSDOC state to the specified source files; and, if *⎕SP* is an empty or numeric vector, catenates the identified source files' tie numbers to *⎕SP*.

QUADFNS
Content option; indicates that system functions are to be included in the WSDOC report's cross-reference tables. This is the default setting. Overrides *NOQUADFNS*.

QUOTES
Content option; indicates that quoted strings to the right of execute functions are to be analysed for cross-reference tables. Overrides *NOQUOTES* (the default).

SAVESRC
File option; indicates that the source file is not to be erased when the WSDOC report is done. Overrides *ERASESRC* (the default).

SET MARGINS [TOP **j**] [*BOTTOM* **k**] [*LEFT* **m**] [*RIGHT* **n**]
Format option; specifies that the top and bottom margins of the WSDOC report are to be **j** and **k** lines deep, respectively; and that the left and right margins are to be **m** and **n** characters wide, respectively. Default: *SET MARGINS TOP* 2 *BOTTOM* 4 *LEFT* 0 *RIGHT* 0.

STAMP
Format option; indicates that a timestamp is to be displayed with each object definition. Overrides *NOSTAMP* (the default).

STATE
State management function; displays the current state.

SURROGATES {'[B][C][I][L][N]'|NONE}
Content option; specifies which non-typeable characters are to be displayed as surrogates in variable and package definitions. Default: *SURROGATES 'BILN'*.

[CENTRE|CENTER] TITLE {BLANK|'**string**'*}*
Format option; specifies a title that is to appear on every page of the WSDOC report. Default: *CENTRE TITLE BLANK*.

TREEHEIGHT **n**
Format option; specifies that **n** is the maximum number of lines which a subtree may occupy in the tree graph structure diagram. Default: *TREEHEIGHT* 10.

TREEROOTS {FINDALL|'**names**'*}*
Content option; specifies that **names** are to be the only functions displayed as root functions in the WSDOC report's tree graph structure diagram. Default: *TREEROOTS FINDALL*.

TTASK
Control option; indicates that the WSDOC job is to run in your current terminal session; that is, in your current T-task. This is the default setting. Overrides *NTASK* or *BTASK*.

WIDTH **n**
Format option; specifies that the WSDOC report's pages are to be **n** characters wide. Default: *WIDTH* 132.

WSDOC
Loads the *WSDOC* workspace, and either begins processing the source files for which you have performed *PUTSTATE*, or else prompts for their tie numbers.

WSDOCSTATUS
Utility program; displays the status of any N- or B-task WSDOC jobs that you've started or requested, or that have ended since the last time you executed *WSDOCSTATUS*.

{*NO*|*NONDEFAULT*|*FULL*} *WSENV*
Content option; specifies the type of workspace environment display that is to appear in the WSDOC report. Default: *NONDEFAULT WSENV*.

WSTOFILE
Creates a WSDOC source file and appends an image of the active workspace to it.

XREFSYMBS {'**symbols**'|*NONE*}
Content option; specifies that **symbols** are to be included in the WSDOC report's cross-reference tables. Default: *XREFSYMBS* '→⍎:⎕⍞'.

WSTOFILE
Utility program; is a commented version of *WSTOFILE*.

Appendix D

WSDOC GLOSSARY

This appendix defines terms peculiar to WSDOC.

branch function
A called function whose subtree is broken out to form its own tree.

cross-reference
Indicates for an identifier: what its class is; where it is used; how it is used; and whether its use is suspicious.

definition
The display of an object's value.

file, output
See *output file.*

file, source
See *source file.*

file name
A file's unique identification. Canonically, a file name is a 22-character vector, the first 10 characters of which contain the library number (left-padded with blanks) and the last 11 characters of which contain the name proper. However, the library number is often omitted if it can be inferred from context; for example, if it is the file user's signon number.

graph
Shows, in the form of a tree, the function-calling hierarchy of the workspace.

identifier
A name used in a workspace, whether or not that name has a value associated with it.

name
The name portion of an *object.*

name, file
See *file name.*

name, workspace
See *workspace name.*

namelist
An array of names to be processed. This array may be: a character scalar, which specifies one one-character name; a character vector, which specifies any number of names, each name separated from the next by a blank or comma; or a character matrix, in which each row contains one name.

object
A name within a workspace, and that name's referent; that is, its associated value; a function, variable, group, or package.

output file
A file containing a WSDOC report.

pathname
Identification of a packed object; or the path to the object. For example, *FFT.TRANSFORMS.MATH* is a pathname; it indicates that *FFT* is packed in *TRANSFORMS*, which in turn is packed in *MATH*. There is no inherent limit to the number of the levels that a pathname may contain.

report
WSDOC's output, which gives the information derived from its analysis of the source file.

root function
The function at the base of each tree.

source file
A file containing the workspace image that WSDOC analyses, usually created by *WSTOFILE*.

state
The collective name for the parameters that determine how WSDOC analyses and then reports on a source file.

summary
In WSDOC, a table giving the names, sizes, and other attributes of all workspace objects of a given class: function, variable, group, or package. By default, WSDOC produces function and variable summaries.

suspicious reference
A use of an identifier which is inconsistent with the identifier's class; for example, localizing an identifier twice. For more information, see "The *FNXREF* Option" and "The *WSXREF* Option".

tree
See *graph*.

workspace name
Uniquely identifies a saved workspace. Canonically, a workspace name is a 22-character vector: the first 10 characters are the library number (left-padded with blanks) and the last 11 characters are the name itself (right-padded with blanks). WSDOC uses the workspace name of the workspace it is documenting to name source and output files.

INDEX

A

ALL option, 25 41
ALPHASORT option, 8 16 **20** 44 45 49

B

BANNERS option, 9 25 29 **30** 46 48 50
branch function, 20 **28** 40 54
BRANCHFNS option, **28** 40 49
BTASK, 5 8 11 **12** 45 47 49

C

content options, 8 24
control options, 8 11
COVER option, 29 **30** 48
CREATEOUT option, 6 8 12 **13** 47 49 51
cross-reference, 54
  tables, 1 9 21 27 28 29 36 37 38 48 51 52 53
CHARS variable, 46 47 **48**
CO variable, 46 47

D

DEFAULT, 9 49
definition, 54
DEFS option, 21 25 **40**
DEPTH option, 8 16 18 49
DESCRIBE variable, 1
DIAMOND option, **22** 47 49

E

effective depth, 16
effective width, 16
ERASESRC option, 8 **13** 47 49

F

file
  output, **2** 8 13 46 47 49 50 55
  source, 1 **2** 9 11 13 21 22 23 30 33 34 35 39 44 45 **46** 47 48 49 50 52 53 **55**
file name, 54
file options, 8 13
format options, 8 16
FNCALLS option, 25 29 **39** 46 48
FNDEF option, 21 23 25 29 **34** 41 47 48
FNS option, 9 21 25 29 **30** 41 48
FNSUM option, 9 25 26 29 **31** 41 48 50
FNSUMC option, 9 25 26 29 **31** 32 41 48 50
FNXREF option, 9 21 22 25 29 **36** 41 46 47 48 50

G

GENWSTOFILE, 44 49
GETSTATE, 10 49
graph, 54
  tree, 20 28 **40**
GRP option, 9 21 25 29 **31** 41 48
GRPS option, 9 21 25 29 **31** 41 48
GRPSUM option, 9 25 29 **33** 41 48

H

HIGHSPEED option, 8 13 **14** 47 50
HSPRINT, 2
  option, 8 13 **14** 47 48 50

I

identifier, 1 21 23 37 46 49 52 **54**
  suspicious, 37 55
  type, 37
  workspace, 46

K

KEYB option, 11 **12** 47 50

L

LIST option, 8 21 24 **25** **29** 48 50
LISTONLY option, 8 24 **25** 29 50

M

MRGWSTOFILES, 44 **45** 50
MSGS option, 25 40 48

N

name, 54
name
  file, see *file name*
  workspace, see *workspace name*
namelist, 42 54
NODIAMOND option, 8 16 **22** 47 50
NOFNDEF option, 8 24 **26** 37 39 48 50
NOHIGHSPEED option, 8 13 **14** 47 50
NOLIST option, 8 24 **25** 29 48 50
NOLISTONLY option, 8 24 **25** 29 50

NONDEFAULT, 24
NONE, 39
  option, 25 **41** 48
NOPKGDEF option, 8 23 24 **27** 37 48 50
NOPRINT option, 5 8 11 **12** 14 40 47 50
NOQUADFNS option, 8 24 **27** 48 51
NOQUOTES option, 8 24 **27** 48 51
NOSTAMP option, 8 16 **22** 47 51
NOVARDEF option, 8 24 **26** 48 51
NTASK, 5 8 **11** 14 45 47 51

O

object, 55
ONLYFNDEF option, 8 24 **26** 37 39 51
ONLYPKGDEF option, 8 24 **27** 37 48 51
ONLYVARDEF option, 8 24 **26** 48 51
OUTFILE option, 6 8 12 **13** 14 49 50 51

P

page format, 16
PAGEFORMAT option, 8 16 **21** 26 51
PAGENOS option, 8 16 **19** 26 51
PAGEPAUSE option, 8 11 **12** 51
pathname, 55
  full, 38
  truncated, 38
PKGDEF option, 9 21 25 29 34 35 **36** 41 47 48
PKGSUM option, 29 **33** 41 48
PRINT option, 4 8 11 **12** 14 47 52
PRINTSUMS option, 8 16 **20** 21 52
PROFID, 42
profile, 4
  default, 3 **42** 45
  STATE, see *STATE profile*
  WSDOC, see *WSDOC profile*
PROFILE, 42
PUTSTATE, 3 **9** 10 52 53

Q

QUADFNS option, 8 24 **27** 48 52
QUOTES option, 8 24 **27** 48 52

R

recursive function, 39 **40**
report, 55
  area, 16
  elements, 9 29
root function, 28 **39** 55

S

SAVESRC option, **13** 47 52
SET MARGINS option, 16 **18** 52
SETPROFILE, 42
STAMP option, 8 16 **22** 47 52
state, 1 55
  changing the state, 9
  current, 3 **7** 49 52
  default, 1 3 **9** 46 49
  options, 8
  WSDOC, 2 7
STATE, 52
  profile, 3 9 42
  program, 7
  workspace, 1 **3** 4 7 44
state-setting functions, 1 2 **3** 9
structure diagram, 39
summary, 55
  function, 29 31
  group, 29 33
  package, 29 33 34
  variable, 29 33
SUMS option, 40
surrogates, 48
SURROGATES option, 28 52
suspicious identifier, 37
suspicious reference, 55
SYSCMDS option, 25 40

T

TBLCONT option, 25 29 **30** 48
TITLE option, 16 **18** 52
tree structure diagram, 1
TREEHEIGHT option, 20 52
TREEROOTS option, 28 52
TTASK, 8 **11** 47 53
type, identifier's, 36
type, object's, 34
type of appearance, 37

V

VARDEF option, 9 21 23 25 29 **34** 41 48
VARS option, 9 21 25 29 **30** 48
VARSUM option, 9 25 29 **33** 48

W

WIDTH option, 8 16 **18** 22 53
workspace name, 55
WSDOC, 1 42 53
  profile, 42

program, 4
workspace, 1 42
WSDOCHOW, 1
WSDOCSTATUS, 3 42 44 **45** 53
WSENV option, 8 **24** 53
WSGRAPH option, 25 29 **39** 46
WSTOFILE, 1 **2** 5 23 42 53 55
WSXREF option, 9 21 25 29 **37** 39 41 46 47 48
WSTOFILE, 3 53

X

XREFS option, 9 21 25 **40**
XREFSYMBS option, 8 24 **28** 36 37 53

Pub. Code: 0349 8409 E1

ISBN 0 86493 094 1

ABBREVIATIONS USED IN A WSDOC

JULY 1981

ABBREVIATION	NAME
AR	Array
BL	Boolean
CH	Character
CM	Complex number
ED	Explicit dyadic function
EM	Explicit monadic function
EN	Explicit niladic function
FL	Floating Point
FN	Function
GV	Global variable
IN	Integer
LA	Left argument
LL	Line label
LV	Local variable
ND	Non-explicit dyadic function
NM	Non-explicit monadic function
NN	Non-explicit niladic function
PK	Package
QF	System quad-function
RA	Right argument
RC	Recursive function
RS	Result

⍎	Locked function
(function name)	Locked function
:	Line label
←	Assigned
I	Indexed
[←	Indexed assignment
*	The variable referenced does not exist locally or globally to the function, or the item being referenced is a special symbol such as a branch arrow (→).
?	Query - The item referenced is suspicious. An item is suspicious if it is: 1) localized but not referenced or assigned in the function, 2) localized and defined as a line-label, 3) localized more than once, 4) defined as a line-label more than once, 5) defined as a line-label and is assigned 6) defined as a line-label and is not referenced, 7) the explicit result of the program and is never assigned a value, 8) referenced but is not defined locally or globally in the workspace.
⎕	Squish-quad. The symbol is unprintable.
B̈	Backspace
Ï	Idle
L̈	Linefeed
N̈	Null
Ü	Uppercase backspace